

日本国特許庁
JAPAN PATENT OFFICE

H. Inoue et al.
3/24/04
Q 80602
10f1

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 3月31日
Date of Application:

出願番号 特願2003-094768
Application Number:
[ST. 10/C]: [JP 2003-094768]

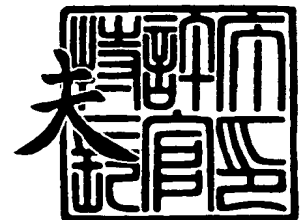
出願人 日本電気株式会社
Applicant(s):



2003年10月 2日

特許庁長官
Commissioner,
Japan Patent Office

今井 康夫



出証番号 出証特2003-3081320

【書類名】 特許願

【整理番号】 34002292

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/16 610

【発明者】

【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

【氏名】 井上 浩明

【発明者】

【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

【氏名】 伊藤 義行

【発明者】

【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

【氏名】 酒井 淳嗣

【発明者】

【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

【氏名】 枝廣 正人

【特許出願人】

【識別番号】 000004237

【氏名又は名称】 日本電気株式会社

【代理人】

【識別番号】 100093595

【弁理士】

【氏名又は名称】 松本 正夫

【手数料の表示】

【予納台帳番号】 057794

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9303563

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 シングルプロセッサ向けOSによる並列処理システム、並列処理プログラム

【特許請求の範囲】

【請求項1】 マルチプロセッサ上でシングルプロセッサ向けOS及びアプリケーションを動作させ、

一のプロセッサ上で動作する前記アプリケーション内で並列化が可能な処理単位を、他のプロセッサ上で新たな処理単位として制御することにより、前記アプリケーションに対して前記マルチプロセッサによる並列処理を行うことを特徴とするシングルプロセッサ向けOSによる並列処理システム。

【請求項2】 マルチプロセッサ上でシングルプロセッサ向けOS及びアプリケーションを動作させ、

前記マルチプロセッサを論理的に、第1のプロセッサ側と第2のプロセッサ側の二つに分け、

前記第1のプロセッサ側のプロセッサ上で動作する前記アプリケーション内で並列化が可能な処理単位を、前記第2のプロセッサ側のプロセッサ上で新たな処理単位として制御することにより、前記アプリケーションに対して前記マルチプロセッサによる並列処理を行うことを特徴とするシングルプロセッサ向けOSによる並列処理システム。

【請求項3】 マルチプロセッサ上でシングルプロセッサ向けOS及びアプリケーションを動作させ、

前記マルチプロセッサを論理的に、第1のプロセッサ側と第2のプロセッサ側の二つに分け、

前記第1のプロセッサ側のプロセッサ上で、前記シングルプロセッサ向けOS及び前記アプリケーションを動作させ、

前記アプリケーション内で並列化が可能な処理単位を、前記第2のプロセッサ側のプロセッサ上で新たな処理単位として制御することにより、前記アプリケーションに対して前記マルチプロセッサによる並列処理を行うことを特徴とするシングルプロセッサ向けOSによる並列処理システム。

【請求項 4】 前記アプリケーション内で並列化が可能な処理単位を、前記第 2 のプロセッサ側のプロセッサ上に予め生成することを特徴とする請求項 1 から請求項 3 の何れか 1 項に記載のシングルプロセッサ向け OS による並列処理システム。

【請求項 5】 前記アプリケーション内で並列化が可能な処理単位を、前記第 2 のプロセッサ側のプロセッサ上で新たな処理単位として生成し起動すること
を特徴とする請求項 1 から請求項 3 の何れか 1 項に記載のシングルプロセッサ向け OS による並列処理システム。

【請求項 6】 前記第 1 プロセッサ側のプロセッサ及び前記第 2 プロセッサ側の各プロセッサに、仮想記憶機構を備える前記シングルプロセッサ向け OS を搭載することを特徴とする請求項 2 から請求項 5 の何れか 1 項に記載のシングルプロセッサ向け OS による並列処理システム。

【請求項 7】 前記第 2 プロセッサ側のプロセッサ上の新たな処理単位を、前記第 1 プロセッサ側のプロセッサ上の処理単位と同期的又は非同期的に制御することを特徴とする請求項 6 に記載のシングルプロセッサ向け OS による並列処理システム。

【請求項 8】 前記第 1 プロセッサ側のプロセッサ及び前記第 2 プロセッサ側のプロセッサ上の処理単位間における、同期的な処理及びデータ受け渡しを可能とすることを特徴とする請求項 6 に記載のシングルプロセッサ向け OS による並列処理システム。

【請求項 9】 前記処理単位間における同期的な処理及びデータ受け渡しをセマフォ方式及びメッセージキュー方式により実行するプロセス間通信手段を、前記第 1 プロセッサ側のプロセッサ及び前記第 2 プロセッサ側のプロセッサに備えることを特徴とする請求項 8 に記載のシングルプロセッサ向け OS による並列処理システム。

【請求項 10】 前記第 1 のプロセッサ側と第 2 のプロセッサ側のそれぞれに、前記処理単位の生成を含む処理単位に関する制御を行う並列処理手段と、前記処理単位に対して前記シングルプロセッサ向け OS のサービスを提供する OS サービス手段を組み込むことを特徴とする請求項 1 から請求項 5 の何れか 1 項に

記載のシングルプロセッサ向けOSによる並列処理システム。

【請求項 11】 前記第1プロセッサ側と第2プロセッサ側間で、制御信号やデータのやり取りを行うための制御処理中継手段を設けたことを特徴とする請求項1から請求項10の何れか1項に記載のシングルプロセッサ向けOSによる並列処理システム。

【請求項 12】 前記制御処理中継手段は、各プロセッサに対応した割り込み制御装置と、各プロセッサに対応した通信領域とを備え、

前記割り込み装置は、他のプロセッサに対して割り込みを指示する割込指示部、割り込みの指示により割り込みがあった旨の情報を保持する割込状態保持部、割り込みをクリアする割込取消部とで構成され、

前記通信領域は、通信元のプロセッサからの通信理由を保持する通信理由保持領域、通信するための通信データを保持する通信データ保持領域、通信を確保するために通信領域に対してロックをかけるための排他制御領域とで構成されることを特徴とする請求項11に記載のシングルプロセッサ向けOSによる並列処理システム。

【請求項 13】 前記制御処理中継手段は、各プロセッサに対応した割り込み制御装置と、各プロセッサに対応した通信領域とを備え、

前記割り込み装置は、他のプロセッサに対して割り込みを指示する割込指示部、割り込みの指示により割り込みがあった旨の情報を保持する割込状態保持部、割り込みをクリアする割込取消部とで構成され、

前記通信領域は、通信元のプロセッサからの通信理由と通信するための通信データを保持する通信キュー、通信を確保するために通信領域に対してロックをかけるための排他制御領域とで構成されることを特徴とする請求項11に記載のシングルプロセッサ向けOSによる並列処理システム。

【請求項 14】 前記第1プロセッサ側に、前記第2プロセッサ側の処理単位と前記シングルプロセッサ向けOS間における各種制御信号の通知を行うため、前記第2プロセッサ側の処理単位と処理単位の番号によって関連付けられ代理手段を設けたことを特徴とする請求項10から請求項12の何れか1項に記載のシングルプロセッサ向けOSによる並列処理システム。

【請求項 15】 前記並列処理手段、OS サービス手段、制御処理中継手段、代理手段をモジュール化して組み込むことを特徴とする請求項 10 から請求項 14 の何れか 1 項に記載のシングルプロセッサ向け OS による並列処理システム。

【請求項 16】 前記第 2 プロセッサ側の前記並列処理手段に、第 2 プロセッサ側の他のプロセッサに対して並列処理させる処理単位を生成する機能を設けたことを特徴とする請求項 10 から請求項 15 の何れか 1 項に記載のシングルプロセッサ向け OS による並列処理システム。

【請求項 17】 前記各プロセッサに、仮想記憶機構を備える前記シングルプロセッサ向け OS を搭載し、前記一のプロセッサ及び前記他のプロセッサ上の処理単位間における、同期的な処理及びデータ受け渡しを可能とすることを特徴とする請求項 1 に記載のシングルプロセッサ向け OS による並列処理システム。

【請求項 18】 前記処理単位間における同期的な処理及びデータ受け渡しをセマフォ方式及びメッセージキュー方式により実行するプロセス間通信手段を、前記各プロセッサに備えることを特徴とする請求項 17 に記載のシングルプロセッサ向け OS による並列処理システム。

【請求項 19】 前記各プロセッサ間相互の前記処理単位間における同期的な処理及びデータ受け渡しに際し、制御信号やデータのやり取りを行うための制御処理中継手段を設けたことを特徴とする請求項 17 又は請求項 18 に記載のシングルプロセッサ向け OS による並列処理システム。

【請求項 20】 マルチプロセッサ上でシングルプロセッサ向け OS 及びアプリケーションを動作させ、

一のプロセッサ上で動作する前記アプリケーション内で並列化が可能な処理単位を、他のプロセッサ上で新たな処理単位として制御することにより、前記アプリケーションに対して前記マルチプロセッサによる並列処理を行う機能を実行することを特徴とするシングルプロセッサ向け OS による並列処理プログラム。

【請求項 21】 マルチプロセッサ上でシングルプロセッサ向け OS 及びアプリケーションを動作させ、

前記マルチプロセッサを論理的に、第 1 のプロセッサ側と第 2 のプロセッサ側

の二つに分けたシステム上で、

前記第1のプロセッサ側のプロセッサ上で動作する前記アプリケーション内で並列化が可能な処理単位を、前記第2のプロセッサ側のプロセッサ上で新たな処理単位として制御することにより、前記アプリケーションに対して前記マルチプロセッサによる並列処理を行う機能を実行することを特徴とするシングルプロセッサ向けOSによる並列処理プログラム。

【請求項22】 マルチプロセッサ上でシングルプロセッサ向けOS及びアプリケーションを動作させ、

前記マルチプロセッサを論理的に、第1のプロセッサ側と第2のプロセッサ側の二つに分けたシステム上で、

前記第1のプロセッサ側のプロセッサ上で、前記シングルプロセッサ向けOS及び前記アプリケーションを動作させ、

前記アプリケーション内で並列化が可能な処理単位を、前記第2のプロセッサ側のプロセッサ上で新たな処理単位として制御することにより、前記アプリケーションに対して前記マルチプロセッサによる並列処理を行う機能を実行することを特徴とするシングルプロセッサ向けOSによる並列処理プログラム。

【請求項23】 前記アプリケーション内で並列化が可能な処理単位を、前記第2のプロセッサ側のプロセッサ上に予め生成する機能を有することを特徴とする請求項20から請求項22の何れか1項に記載のシングルプロセッサ向けOSによる並列処理プログラム。

【請求項24】 前記アプリケーション内で並列化が可能な処理単位を、前記第2のプロセッサ側のプロセッサ上で新たな処理単位として生成し起動する機能を有することを特徴とする請求項20から請求項22の何れか1項に記載のシングルプロセッサ向けOSによる並列処理プログラム。

【請求項25】 仮想記憶機構を備える前記シングルプロセッサ向けOSを搭載した前記第1プロセッサ側のプロセッサ及び前記第2プロセッサ側の各プロセッサ上で実行されることを特徴とする請求項20から請求項24の何れか1項に記載のシングルプロセッサ向けOSによる並列処理プログラム。

【請求項26】 前記第2プロセッサ側のプロセッサ上の新たな処理単位を

、前記第 1 プロセッサ側のプロセッサ上の処理単位と同期的又は非同期的に制御する機能を有することを特徴とする請求項 25 に記載のシングルプロセッサ向け OS による並列処理プログラム。

【請求項 27】 前記第 1 プロセッサ側のプロセッサ及び前記第 2 プロセッサ側のプロセッサ上の処理単位間における、同期的な処理及びデータ受け渡しを可能とする機能を有することを特徴とする請求項 25 に記載のシングルプロセッサ向け OS による並列処理プログラム。

【請求項 28】 前記処理単位間における同期的な処理及びデータ受け渡しをセマフォ方式及びメッセージキュー方式により実行するプロセス間通機能を、前記第 1 プロセッサ側のプロセッサ及び前記第 2 プロセッサ側のプロセッサで実行することを特徴とする請求項 27 に記載のシングルプロセッサ向け OS による並列処理プログラム。

【請求項 29】 前記第 1 のプロセッサ側と第 2 のプロセッサ側のそれぞれに、前記処理単位の生成を含む処理単位に関する制御を行う並列処理機能と、前記処理単位に対して前記シングルプロセッサ向け OS のサービスを提供する OS サービス機能を持たせることを特徴とする請求項 20 から請求項 24 の何れか 1 項に記載のシングルプロセッサ向け OS による並列処理プログラム。

【請求項 30】 前記第 1 プロセッサ側と第 2 プロセッサ側間で、制御信号やデータのやり取りを行うための制御処理中継機能を有することを特徴とする請求項 20 から請求項 29 の何れか 1 項に記載のシングルプロセッサ向け OS による並列処理プログラム。

【請求項 31】 前記第 1 プロセッサ側に、前記第 2 プロセッサ側の処理単位と前記シングルプロセッサ向け OS 間における各種制御信号の通知を行うため、前記第 2 プロセッサ側の処理単位と処理単位の番号によって関連付けられ代理機能を有することを特徴とする請求項 29 又は請求項 30 に記載のシングルプロセッサ向け OS による並列処理プログラム。

【請求項 32】 前記第 2 プロセッサ側の前記並列処理機能が、第 2 プロセッサ側の他のプロセッサに対して並列処理させる処理単位を生成することを特徴とする請求項 29 から請求項 31 の何れか 1 項に記載のシングルプロセッサ向け

OS による並列処理プログラム。

【請求項 3 3】 仮想記憶機構を備える前記シングルプロセッサ向け OS を搭載した前記各プロセッサ上で、前記一のプロセッサ及び前記他のプロセッサ上の処理単位間における、同期的な処理及びデータ受け渡しを可能とすることを特徴とする請求項 2 0 に記載のシングルプロセッサ向け OS による並列処理プログラム。

【請求項 3 4】 前記処理単位間における同期的な処理及びデータ受け渡しをセマフォ方式及びメッセージキュー方式により実行するプロセス間通信機能を用いて、前記各プロセッサ上で実行することを特徴とする請求項 3 3 に記載のシングルプロセッサ向け OS による並列処理プログラム。

【請求項 3 5】 前記各プロセッサ間相互の前記処理単位間における同期的な処理及びデータ受け渡しに際し、制御信号やデータのやり取りを行うための制御処理中継機能を実行することを特徴とする請求項 3 3 又は請求項 3 4 に記載のシングルプロセッサ向け OS による並列処理プログラム。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、マルチプロセッサによる並列処理システムに関し、特に、マルチプロセッサ上でシングルプロセッサ向け OS 及び既存のアプリケーションを動作させ、アプリケーションに対してマルチプロセッサによる並列処理を実現するシングルプロセッサ向け OS による並列処理システム及び並列処理プログラムに関する。

【0 0 0 2】

【従来の技術】

携帯電話及び携帯 PC を含む携帯端末等のデータ処理装置においては、基本的にシングルプロセッサ上でシングルプロセッサ向けオペレーティングシステム（以下、シングルプロセッサ向け OS）及びシングルプロセッサ向けのアプリケーション（以下、単にアプリケーションと称する）を実行させているのが現状である。

【0003】

このような状況において、マルチプロセッサベースで上記アプリケーションそのままを使用したい場合には、上記シングルプロセッサ向けOSの代わりにマルチプロセッサ向けOS上でアプリケーションを実行させる必要がある。

【0004】

上記のようなマルチプロセッサシステム上でマルチプロセッサ向けOSとアプリケーションの実行を制御するシステムとしては、例えば、特開平3-257652号公報、特開平3-113563号公報に開示される従来技術がある。

【0005】

特開平3-257652号公報（特許文献1）では、複数のプロセッサエレメントからなるマルチプロセッサシステムにおける各プロセッサエレメント間の割り込み制御方法を開示する。

【0006】

特開平3-113563号公報（特許文献2）では、マルチプロセッサシステムにおける複数のプロセッサに割り当てるプロセスのスケジューリング方法を開示する。

【0007】

また、特開2003-058515号公報（特許文献3）では、複数のプロセッサエレメントで個別のプロセスを実行するための方法を開示している。

【0008】**【特許文献1】**

特開平3-257652号公報

【特許文献2】

特開平3-113563号公報

【特許文献3】

特開2003-058515号公報

【0009】**【発明が解決しようとする課題】**

しかし、従来のように、マルチプロセッサ向けOS上で既存のアプリケーション

ンを動作させる場合、アプリケーションは複数のプロセッサのうちの一つしか使用しないにもかかわらず、マルチプロセッサ向けOSがマルチプロセッサ向けのサービスを行ったり、他のアプリケーションが動作していないのに他のプロセッサとの排他処理を続行するために、その余分な処理がオーバーヘッドになってしまふという問題があった。

【0010】

また、上記アプリケーションをマルチプロセッサ用に改造すれば上記問題点が解消するが、改造には非常に手間とコストがかかるという別の問題点が発生する。

【0011】

また、アプリケーションがそのまま既存のシングルプロセッサ向けOS上で利用されている場合には、既存のアプリケーションをそのままマルチプロセッサ上で動作させたいという要求が高い。

【0012】

更に、並列プロセッサにおけるプロセス間の通信には、通信用データを共有する機能と、プロセスのスリープとウェイクアップを行なう機構とが必要であるが、従来では上記機能をOSのカーネルレベルで実装することで実現していた。すなわち、並列プロセッサでのプロセス間の通信には並列プロセッサ向けのOSを必要としていた。ここで、プロセス間の通信とは、プロセス間相互の処理の同期とデータ受け渡しを意味する。

【0013】

従って、上記のようにマルチプロセッサの各プロセッサ上でシングルプロセッサ向けOSを動作させるシングルプロセッサ向けOSの単なる並列化では、同じプロセッサ内のプロセス同士の通信はシングルプロセッサ向けOSの機能により可能となるが、異なるプロセッサ相互のプロセス間での通信はできないという問題がある。

【0014】

例えば、特許文献3においては、異なるプロセッサ相互のプロセス間通信を行なう方法については何ら開示されていない。

【0015】

なお、シングルプロセッサ向けOS本来の機能としてではなく、例えばネットワークを利用することにより、異なるプロセッサ相互のプロセス間での通信が可能であるが、ネットワーク通信の処理の遅さがオーバーヘッドとなって、マルチプロセッサによる並列処理を行う意義が失われてしまう。

【0016】

本発明の目的は、マルチプロセッサ上でシングルプロセッサ向けOSと既存のアプリケーションをそれらに改造を加えることなく動作させ、その既存のアプリケーションに対してマルチプロセッサによる並列処理を実現できるようにしたシングルプロセッサ向けOSによる並列処理システム及び並列処理プログラムを提供することにある。

【0017】

本発明の他の目的は、マルチプロセッサ上でシングルプロセッサ向けOSと既存のアプリケーションをそれらに改造を加えることなく動作させると共に、各プロセッサ間でアプリケーションの処理単位を、同期的又は非同期的に制御することを可能とするシングルプロセッサ向けOSによる並列処理システム及び並列処理プログラムを提供することにある。

【0018】

本発明のさらに他の目的は、ユーザレベルでプロセス間の通信機能を実装することにより、マルチプロセッサ向けOSを必要とすることなく、シングルプロセッサ向けOS上でも、並列プロセッサでのプロセス間の通信を可能とするシングルプロセッサ向けOSによる並列処理システム及び並列処理プログラムを提供することにある。

【0019】**【課題を解決するための手段】**

上記目的を達成するため本発明のシングルプロセッサ向けOSによる並列処理システムは、マルチプロセッサ上でシングルプロセッサ向けOS及びアプリケーションを動作させ、一のプロセッサ上で動作する前記アプリケーション内で並列化が可能な処理単位を、他のプロセッサ上で新たな処理単位として制御すること

により、前記アプリケーションに対して前記マルチプロセッサによる並列処理を行うことを特徴とする。

【 0 0 2 0 】

請求項 2 のシングルプロセッサ向け OS による並列処理システムは、マルチプロセッサ上でシングルプロセッサ向け OS 及びアプリケーションを動作させ、前記マルチプロセッサを論理的に、第 1 のプロセッサ側と第 2 のプロセッサ側の二つに分け、前記第 1 のプロセッサ側のプロセッサ上で動作する前記アプリケーション内で並列化が可能な処理単位を、前記第 2 のプロセッサ側のプロセッサ上で新たな処理単位として制御することにより、前記アプリケーションに対して前記マルチプロセッサによる並列処理を行うことを特徴とする。

【 0 0 2 1 】

請求項 3 のシングルプロセッサ向け OS による並列処理システムは、マルチプロセッサ上でシングルプロセッサ向け OS 及びアプリケーションを動作させ、前記マルチプロセッサを論理的に、第 1 のプロセッサ側と第 2 のプロセッサ側の二つに分け、前記第 1 のプロセッサ側のプロセッサ上で、前記シングルプロセッサ向け OS 及び前記アプリケーションを動作させ、前記アプリケーション内で並列化が可能な処理単位を、前記第 2 のプロセッサ側のプロセッサ上で新たな処理単位として制御することにより、前記アプリケーションに対して前記マルチプロセッサによる並列処理を行うことを特徴とする。

【 0 0 2 2 】

請求項 4 のシングルプロセッサ向け OS による並列処理システムは、前記アプリケーション内で並列化が可能な処理単位を、前記第 2 のプロセッサ側のプロセッサ上に予め生成することを特徴とする。

【 0 0 2 3 】

請求項 5 のシングルプロセッサ向け OS による並列処理システムは、前記アプリケーション内で並列化が可能な処理単位を、前記第 2 のプロセッサ側のプロセッサ上で新たな処理単位として生成し起動することを特徴とする。

【 0 0 2 4 】

請求項 6 のシングルプロセッサ向け OS による並列処理システムは、前記第 1

プロセッサ側のプロセッサ及び前記第 2 プロセッサ側の各プロセッサに、仮想記憶機構を備える前記シングルプロセッサ向け OS を搭載することを特徴とする。

【 0 0 2 5 】

請求項 7 のシングルプロセッサ向け OS による並列処理システムは、前記第 2 プロセッサ側のプロセッサ上の新たな処理単位を、前記第 1 プロセッサ側のプロセッサ上の処理単位と同期的又は非同期的に制御することを特徴とする。

【 0 0 2 6 】

請求項 8 のシングルプロセッサ向け OS による並列処理システムは、前記第 1 プロセッサ側のプロセッサ及び前記第 2 プロセッサ側のプロセッサ上の処理単位間における、同期的な処理及びデータ受け渡しを可能とすることを特徴とする。

【 0 0 2 7 】

請求項 9 のシングルプロセッサ向け OS による並列処理システムは、前記処理単位間における同期的な処理及びデータ受け渡しをセマフォ方式及びメッセージキュー方式により実行するプロセス間通信手段を、前記第 1 プロセッサ側のプロセッサ及び前記第 2 プロセッサ側のプロセッサに備えることを特徴とする。

【 0 0 2 8 】

請求項 1 0 のシングルプロセッサ向け OS による並列処理システムは、前記第 1 のプロセッサ側と第 2 のプロセッサ側のそれぞれに、前記処理単位の生成を含む処理単位に関する制御を行う並列処理手段と、前記処理単位に対して前記シングルプロセッサ向け OS のサービスを提供する OS サービス手段を組み込むことを特徴とする。

【 0 0 2 9 】

請求項 1 1 のシングルプロセッサ向け OS による並列処理システムは、前記第 1 プロセッサ側と第 2 プロセッサ側間で、制御信号やデータのやり取りを行うための制御処理中継手段を設けたことを特徴とする。

【 0 0 3 0 】

請求項 1 2 のシングルプロセッサ向け OS による並列処理システムは、前記制御処理中継手段は、各プロセッサに対応した割り込み制御装置と、各プロセッサに対応した通信領域とを備え、前記割り込み装置は、他のプロセッサに対して割

り込みを指示する割込指示部、割り込みの指示により割り込みがあった旨の情報を保持する割込状態保持部、割り込みをクリアする割込取消部とで構成され、前記通信領域は、通信元のプロセッサからの通信理由を保持する通信理由保持領域、通信するための通信データを保持する通信データ保持領域、通信を確保するために通信領域に対してロックをかけるための排他制御領域とで構成されることを特徴とする。

【0031】

請求項13のシングルプロセッサ向けOSによる並列処理システムは、前記制御処理中継手段は、各プロセッサに対応した割り込み制御装置と、各プロセッサに対応した通信領域とを備え、前記割り込み装置は、他のプロセッサに対して割り込みを指示する割込指示部、割り込みの指示により割り込みがあった旨の情報を保持する割込状態保持部、割り込みをクリアする割込取消部とで構成され、前記通信領域は、通信元のプロセッサからの通信理由と通信するための通信データを保持する通信キュー、通信を確保するために通信領域に対してロックをかけるための排他制御領域とで構成されることを特徴とする。

【0032】

請求項14のシングルプロセッサ向けOSによる並列処理システムは、前記第1プロセッサ側に、前記第2プロセッサ側の処理単位と前記シングルプロセッサ向けOS間における各種制御信号の通知を行うため、前記第2プロセッサ側の処理単位と処理単位の番号によって関連付けられ代理手段を設けたことを特徴とする。

【0033】

請求項15のシングルプロセッサ向けOSによる並列処理システムは、前記並列処理手段、OSサービス手段、制御処理中継手段、代理手段をモジュール化して組み込むことを特徴とする。

【0034】

請求項16のシングルプロセッサ向けOSによる並列処理システムは、前記第2プロセッサ側の前記並列処理手段に、第2プロセッサ側の他のプロセッサに対して並列処理させる処理単位を生成する機能を設けたことを特徴とする。

【 0 0 3 5 】

請求項 1 7 のシングルプロセッサ向け OS による並列処理システムは、前記各プロセッサに、仮想記憶機構を備える前記シングルプロセッサ向け OS を搭載し、前記一のプロセッサ及び前記他のプロセッサ上の処理単位間における、同期的な処理及びデータ受け渡しを可能とすることを特徴とする。

【 0 0 3 6 】

請求項 1 8 のシングルプロセッサ向け OS による並列処理システムは、前記処理単位間における同期的な処理及びデータ受け渡しをセマフォ方式及びメッセージキュー方式により実行するプロセス間通信手段を、前記各プロセッサに備えることを特徴とする。

【 0 0 3 7 】

請求項 1 9 のシングルプロセッサ向け OS による並列処理システムは、マルチプロセッサ上でシングルプロセッサ向け OS 及びアプリケーションを動作させ、一のプロセッサ上で動作する前記アプリケーション内で並列化が可能な処理単位を、他のプロセッサ上で新たな処理単位として制御することにより、前記アプリケーションに対して前記マルチプロセッサによる並列処理を行う機能を実行することを特徴とする。

【 0 0 3 8 】

請求項 2 0 のシングルプロセッサ向け OS による並列処理プログラムは、マルチプロセッサ上でシングルプロセッサ向け OS 及びアプリケーションを動作させ、前記マルチプロセッサを論理的に、第 1 のプロセッサ側と第 2 のプロセッサ側の二つに分けたシステム上で、前記第 1 のプロセッサ側のプロセッサ上で動作する前記アプリケーション内で並列化が可能な処理単位を、前記第 2 のプロセッサ側のプロセッサ上で新たな処理単位として制御することにより、前記アプリケーションに対して前記マルチプロセッサによる並列処理を行う機能を実行することを特徴とする。

【 0 0 3 9 】

請求項 2 1 のシングルプロセッサ向け OS による並列処理プログラムは、マルチプロセッサ上でシングルプロセッサ向け OS 及びアプリケーションを動作させ

、前記マルチプロセッサを論理的に、第1のプロセッサ側と第2のプロセッサ側の二つに分けたシステム上で、前記第1のプロセッサ側のプロセッサ上で、前記シングルプロセッサ向けOS及び前記アプリケーションを動作させ、前記アプリケーション内で並列化が可能な処理単位を、前記第2のプロセッサ側のプロセッサ上で新たな処理単位として制御することにより、前記アプリケーションに対して前記マルチプロセッサによる並列処理を行う機能を実行することを特徴とする。

【0040】

請求項22のシングルプロセッサ向けOSによる並列処理プログラムは、マルチプロセッサ上でシングルプロセッサ向けOS及びアプリケーションを動作させ、前記マルチプロセッサを論理的に、第1のプロセッサ側と第2のプロセッサ側の二つに分けたシステム上で、前記第1のプロセッサ側のプロセッサ上で、前記シングルプロセッサ向けOS及び前記アプリケーションを動作させ、前記アプリケーション内で並列化が可能な処理単位を、前記第2のプロセッサ側のプロセッサ上で新たな処理単位として制御することにより、前記アプリケーションに対して前記マルチプロセッサによる並列処理を行う機能を実行することを特徴とする。

【0041】

請求項23のシングルプロセッサ向けOSによる並列処理プログラムは、前記アプリケーション内で並列化が可能な処理単位を、前記第2のプロセッサ側のプロセッサ上に予め生成する機能を有することを特徴とする。

【0042】

請求項24のシングルプロセッサ向けOSによる並列処理プログラムは、前記アプリケーション内で並列化が可能な処理単位を、前記第2のプロセッサ側のプロセッサ上で新たな処理単位として生成し起動する機能を有することを特徴とする。

【0043】

請求項25のシングルプロセッサ向けOSによる並列処理プログラムは、仮想記憶機構を備える前記シングルプロセッサ向けOSを搭載した前記第1プロセッサ

サ側のプロセッサ及び前記第2プロセッサ側の各プロセッサ上で実行されることを特徴とする。

【0044】

請求項26のシングルプロセッサ向けOSによる並列処理プログラムは、前記第2プロセッサ側のプロセッサ上の新たな処理単位を、前記第1プロセッサ側のプロセッサ上の処理単位と同期的又は非同期的に制御する機能を有することを特徴とする。

【0045】

請求項27のシングルプロセッサ向けOSによる並列処理プログラムは、前記第1プロセッサ側のプロセッサ及び前記第2プロセッサ側のプロセッサ上の処理単位間における、同期的な処理及びデータ受け渡しを可能とする機能を有することを特徴とする。

【0046】

請求項28のシングルプロセッサ向けOSによる並列処理プログラムは、前記処理単位間における同期的な処理及びデータ受け渡しをセマフォ方式及びメッセージキュー方式により実行するプロセス間通機能を、前記第1プロセッサ側のプロセッサ及び前記第2プロセッサ側のプロセッサで実行することを特徴とする。

【0047】

請求項29のシングルプロセッサ向けOSによる並列処理プログラムは、前記第1のプロセッサ側と第2のプロセッサ側のそれぞれに、前記処理単位の生成を含む処理単位に関する制御を行う並列処理機能と、前記処理単位に対して前記シングルプロセッサ向けOSのサービスを提供するOSサービス機能を持たせることを特徴とする。

【0048】

請求項30のシングルプロセッサ向けOSによる並列処理プログラムは、前記第1プロセッサ側と第2プロセッサ側間で、制御信号やデータのやり取りを行うための制御処理中継機能を有することを特徴とする。

【0049】

請求項31のシングルプロセッサ向けOSによる並列処理プログラムは、前記

第1プロセッサ側に、前記第2プロセッサ側の処理単位と前記シングルプロセッサ向けOS間における各種制御信号の通知を行うため、前記第2プロセッサ側の処理単位と処理単位の番号によって関連付けられ代理機能を有することを特徴とする。

【0050】

請求項32のシングルプロセッサ向けOSによる並列処理プログラムは、前記第2プロセッサ側の前記並列処理機能が、第2プロセッサ側の他のプロセッサに対して並列処理させる処理単位を生成することを特徴とする。

【0051】

請求項33のシングルプロセッサ向けOSによる並列処理プログラムは、仮想記憶機構を備える前記シングルプロセッサ向けOSを搭載した前記各プロセッサ上で、前記一のプロセッサ及び前記他のプロセッサ上の処理単位間における、同期的な処理及びデータ受け渡しを可能とすることを特徴とする。

【0052】

請求項34のシングルプロセッサ向けOSによる並列処理プログラムは、前記処理単位間における同期的な処理及びデータ受け渡しをセマフォ方式及びメッセージキュー方式により実行するプロセス間通信機能を、前記各プロセッサ上で実行することを特徴とする。

【0053】

請求項35のシングルプロセッサ向けOSによる並列処理プログラムは、前記各プロセッサ間相互の前記処理単位間における同期的な処理及びデータ受け渡しに際し、制御信号やデータのやり取りを行うための制御処理中継機能を実行することを特徴とする。

【0054】

【発明の実施の形態】

本発明の並列処理システムでは、従来のシングルプロセッサ向けOSに対して一切の改造を加えることなく、複数のプロセッサへの処理を依頼する機構や複数のプロセッサへのOSサービスの提供におけるクリティカルセクションの保護機構を付加することにより、マルチプロセッサ上での並列処理を実行する。

【0055】

以下本発明の好適な実施の形態について図面を参照して詳細に説明する。

【0056】

図1は本発明の好適な実施の形態による並列処理システムの構成を示すブロック図である。

【0057】

図1に示すように、本実施の形態による並列処理システムは、システムバス91で接続される複数のプロセッサ(CPU)P0～Pn(nは1以上の整数)からなるマルチプロセッサを、論理的に第1プロセッサ側10と第2プロセッサ側20の二組に分け、第1プロセッサ側10のプロセッサP0上で動作するシングルプロセッサ向けOS30を実装し、かつ第1プロセッサ側10のプロセッサP0及び第2プロセッサ側20のプロセッサP1～Pnのそれぞれに、並列処理を行うための並列処理手段40P0～40Pn及びOSサービス処理するためのOSサービス手段50P0～50Pnを実装して構成される。また、システムバス91には、各プロセッサP0～Pnが共有する共有メモリとしての主記憶装置92及びディスク装置等の外部記憶装置93が接続される。

【0058】

なお、上記マルチプロセッサには、同一種類のプロセッサを複数備える構成だけでなく、DSPやセキュリティエンジン等の異種の処理装置を複数備える構成も含む。

【0059】

また、第1プロセッサ側10と第2プロセッサ側20間で、制御信号やデータのやり取りを行うための制御処理中継手段60が独自に設けられ、かつ第1プロセッサ側10には、シングルプロセッサ向けOS30が第2プロセッサ側20で実行されるタスクと通信するための代理手段70が設けられている。

【0060】

上記第1プロセッサ側10のプロセッサは、必ずしも単一である必要はなく複数のプロセッサからなってもよい。例えば、第1プロセッサ側10を2つのプロセッサとし、それぞれに異なるシングルプロセッサ向けOSを実装するよう

な構成とすることも可能である。

【0061】

また、第1プロセッサ側10のプロセッサで起動されるシングルプロセッサ向けOS30としては、既存のOSが利用される。例えば、リアルタイムOSやUNIX(R) OSがそのまま利用される。

【0062】

本明細書で使用するタスクとは、UNIX(R) OSにおけるプロセスやスレッド、リアルタイムOSにおけるタスク等の並列処理を行うための処理単位を意味している。

【0063】

本実施の形態による並列処理システムにおいては、第1プロセッサ側10のシングルプロセッサ向けOS上でアプリケーションが動作し、そのアプリケーションの処理単位のうち、並列化を行えないタスク（逐次タスク）を第1プロセッサ側10のプロセッサP0で処理し、アプリケーション内で並列化を行えるタスクを第2プロセッサ側20において新たなタスクとして生成して並列処理をさせる。

【0064】

並列処理手段40P0及び並列処理手段40P1～40Pnは、タスクの生成、起動、停止、終了、削除、及びその他のタスクに関する制御を行う機能を有する。ここでは、第1プロセッサ側10の並列処理手段40P0から第2プロセッサ側20の各プロセッサP1～Pnの並列処理手段40P1～40Pnに対して、制御処理中継手段60を介してタスクの生成、起動、停止、終了、削除等の処理を行う。また、シグナル通知については、並列処理手段40P0と並列処理手段40P1～40Pnの双方向から処理される。

【0065】

OSサービス手段50P0及びOSサービス手段50P1～50Pnは、外部機器への各種のアクセスと制御を行うためのインタフェース、タスクどうしが共有するリソースへの各種アクセスと制御を行うためのインタフェースとして機能を有する。

【0066】

制御処理中継手段60は、第1プロセッサ側10と第2プロセッサ側20間で、制御信号やデータのやり取りを行うための手段であり、複数のプロセッサで並列処理される複数のタスク間の制御に使用される。

【0067】

代理手段70は、第2プロセッサ側20で実行されるタスク（一部あるいは全部）に対応付けられ、第2プロセッサ側20のタスクとシングルプロセッサ向けOS30間におけるシグナル通知（タスクを制御するための各種制御信号の通知）のために実装されている。

【0068】

以下、上記のように構成される本実施の形態による並列処理システムの動作について、図面を参照して詳しく説明する。

【0069】

ここでは、第1プロセッサ側10のシングルプロセッサ向けOS上でアプリケーションが動作し、そのアプリケーションの処理単位のうち、第1プロセッサ側10のプロセッサP0で処理する処理単位を逐次タスクSTとし、アプリケーション内で並列化を行えるタスクであって、第2プロセッサ側20に対してタスクPT-1～PT-nとして並列処理をさせる処理単位を並列化タスクPTとする。

【0070】

まず、並列処理手段40P0、40P1～40Pnによる並列処理起動の動作について、図2と図3を用いて説明する。

【0071】

(1) 第1プロセッサ側10の並列化タスクPTを、第2プロセッサ側20のタスクPT-1～PT-nの何れかとして起動させる場合、並列処理手段40P0に対して、第2プロセッサ側20で起動させる処理単位としてのタスクPT-1～PT-nの何れかの生成を要求する。

【0072】

並列処理手段40P0、並列処理手段40P1～40Pnからの要求（comman

d) としては、例えば、以下のようなものがある。create (タスクの生成)、delete (タスクの削除)、activate (タスクの起動)、terminate (タスクの終了)、signal (シグナル要求) 等がある。このうち、create (タスクの生成)、delete (タスクの削除)、activate (タスクの起動)、terminate (タスクの終了) については、第1プロセッサ側10から第2プロセッサ側20へ送られる要求であり、signal (シグナル要求) は、第1プロセッサ側10と第2プロセッサ側20間で双方向から送られる。

【0073】

この要求は、図3に示すようなメッセージによって行われる。すなわち、要求する内容（この場合は、タスク生成）を示す要求内容201、タスクを処理させる第2プロセッサ側20のプロセッサP1～Pnを指定するプロセッサ番号202、新たに生成するタスクPT-1～PT-nを示すタスク番号、タスク生成を要求する並列化タスクPTを示す元タスク番号からなるメッセージを並列処理手段40P0に対して送信することで、タスク生成の要求を行う。なお、タスク生成以外の要求についても、図3に示すメッセージに従った要求が送られる。

【0074】

(2) これに対して、並列処理手段40P0は、生成するタスクPT-1～PT-nに対応する代理手段70を起動する。この代理手段70は、第1プロセッサ側10と第2プロセッサ側20でタスクの管理内容を共有しないようにするため、かつ排他処理を第1プロセッサ側10で完結させるために起動される。この時、代理手段70にはタスクPT-1～PT-nのタスク番号が保持される。

【0075】

(3) さらに、並列処理手段40P0は、上述したタスク番号及び要求内容（第2プロセッサ側20のプロセッサP1～Pnへのタスク生成）といったタスク生成に必要なデータと、「並列処理」を指定する通信理由情報とを制御処理中継手段60に設定する。

【0076】

これにより、制御処理中継手段60に対して、後述する主記憶装置92（共有メモリ）への通信内容の設定とプロセッサ間割り込みのための処理が行われる。

【0077】

ここで、通信理由情報とは、制御処理中継手段60に渡したデータの受け取り先（取得先）を示すものであり、上記の場合、指定されたプロセッサP1～Pnの並列処理手段40P1～40Pnによって、制御処理中継手段60に設定されたデータが取得されることになる。

【0078】

（4）第2プロセッサ側20の指定されたプロセッサP1～Pn上の並列処理手段40P1～40Pnは、制御処理中継手段60から通信理由情報が「並列処理」の要求内容（タスク制御に必要なデータ）を取得する。

【0079】

（5）そして、並列処理手段40P1～40Pnは、取得した要求内容に基づいてプロセッサP1～Pn上でタスクPT-1～PT-nを生成・起動する。

【0080】

以上により、第1プロセッサ側10のシングルプロセッサ向けOS上で動作するアプリケーションの処理単位である並列化タスクPTの処理単位をを、第2プロセッサ側20上にタスクPT-1～PT-nとして並列処理をさせることが可能となる。

【0081】

なお、上記の動作においては、並列処理手段40P1～40Pnは、取得した要求内容に基づいてプロセッサP1～Pn上でタスクPT-1～PT-nを生成して起動する場合について述べたが、第1プロセッサ側10の並列化タスクPTを、予め第2プロセッサ側20のタスクPT-1～PT-nの何れかとして生成しておき、並列処理手段40P1～40Pnが、取得した要求内容に基づいてプロセッサP1～Pn上でタスクPT-1～PT-nを起動するようにすることもできる。

【0082】

次に、OSサービス手段50P0、50P1～50PnによるOSサービス処理動作について説明する。

【0083】

OSサービス手段50P0、50P1～50Pnは、第2プロセッサ側20のプロセッサP1～Pn上に生成したタスクPT-1～PT-nからの要求に基づいて、シングルプロセッサ向けOS30によるサービスである、外部機器への各種アクセスと制御、他のタスクと共用するリソースへの各種アクセスと制御に関するサービスを提供するための機能を有する。シングルプロセッサ向けOS30によるサービスには、通常のOSが提供するシステムコールやAPIに相当するものが主な対象となる。

【0084】

第2プロセッサ側20のプロセッサP1～Pn上のタスクPT-1～PT-nからのファイルアクセス（例えば、外部記憶装置93上のファイルに対する各種の処理）要求に対するOSサービス手段50P0、50P1～50Pnの動作を図4を参照して説明する。

【0085】

ここで、ファイルアクセスには、open（ファイルのオープン）、close（ファイルのクローズ）、read（ファイルの読み出し）、write（ファイルの書き込み）、seek（ファイル書き込み位置の移動）、remove（ファイルの削除）、rename（ファイルの名称変更）等の処理が含まれる。

【0086】

（1）第2プロセッサ側20のタスクPT-1～PT-nにおいてファイルアクセス処理の必要が発生すると、タスクPT-1～PT-nが、第2プロセッサ側20のOSサービス手段50P1～50Pnに対してファイルアクセスのためのサービスを要求する。このファイルアクセスのサービス要求においては、例えば、OSサービス手段50P1～50Pnのファイルへの書き込み処理として定義されたwrite関数やファイルの読み出し処理として定義されたread関数等が呼び出される。

【0087】

ここで、OSサービス手段50P1～50Pnは、第1プロセッサ側10の処理（シングルプロセッサ向けOS30によるファイルアクセス処理）に必要なデータを設定する。ここで、必要なデータには、要求内容（例えば、write reques

t)、アクセスするファイルの記述子 (file descriptor)、文字列へのポインタ、文字列の長さ、タスク番号等の情報が含まれる。

【0088】

(2) OSサービス手段50P1~50Pnは、通信理由情報を「OSサービス」として、要求内容を含む必要なデータを制御処理中継手段60に対して設定することで、第1プロセッサ側10に対してファイルアクセス要求を発行する。

【0089】

その後、ファイルアクセスのサービス要求を行ったタスクPT-1~PT-nは、待ち状態となり、対応するプロセッサP1~Pnにおいては並列処理手段40P1~40Pnにより他のタスクへ処理が切り替わる (タスクのswitch:スイッチ)。

【0090】

(3) 第1プロセッサ側10のOSサービス手段50P0は、制御処理中継手段60から「OSサービス」という通信理由情報の上記ファイルアクセス要求を取得する。

【0091】

(4) 第1プロセッサ側10のOSサービス手段50P0は、取得した要求内容に従ってシングルプロセッサ向けOS30に対してファイルアクセスを要求する。

【0092】

(5) これにより、シングルプロセッサ向けOS30は、要求に基づいて外部記憶装置93等に対するファイルアクセス (writeやread等) を行う。このファイルアクセス処理は、シングルプロセッサ向けOS30が本来備えているファイルアクセスサービスをそのまま利用して実行される。

【0093】

(6) 要求されたファイルアクセス処理が終了すると、シングルプロセッサ向けOS30は、第1プロセッサ側10のOSサービス手段50P0にファイルアクセス要求に対する戻り値を返し、処理を戻す。

【0094】

(7) さらに、OSサービス手段50P0は、その戻り値とファイルアクセス要求を行ったタスクPT-1～PT-nのタスク番号を含むデータである通信内容を、「OSサービス」を通信理由情報として制御処理中継手段60に設定することにより、第2プロセッサ側20のプロセッサP1～Pnに対してファイルアクセスの完了を通知する。

【0095】

(8) 対応するプロセッサP1～PnのOSサービス手段50P1～50Pnは、制御処理中継手段60から上記設定された戻り値と完了通知を受け取る。

【0096】

(9) そして、第2プロセッサ側20のOSサービス手段50P1～50Pnは、並列処理手段40P1～40Pnに対してファイルアクセス要求を行ったタスクPT-1～PT-nを起動するように依頼する。

【0097】

これにより、待ち状態となっていたタスクPT-1～PT-nに処理が切り替わる。

【0098】

(10) 並列処理手段40P1～40Pnから起動されたタスクPT-1～PT-nは、OSサービス手段50P1～50Pnからファイルアクセスの戻り値を受け取り、処理を継続する。

【0099】

以上により、第2プロセッサ側20に個別にファイルアクセスのための処理手段を設けることなく、第2プロセッサ側20におけるタスクPT-1～PT-nはシングルプロセッサ向けOS30のサービスをそのまま利用してファイルアクセスが可能となる。また、これにより、ファイルアクセスのための排他処理については、第1プロセッサ側10で完結するため、マルチプロセッサ向けOS上でアプリケーションを動作させる場合のような排他処理によるオーバーヘッドが発生することなく並列処理を実現できる。

【0100】

なお、第2プロセッサ側20のプロセッサP1～Pn上のタスクPT-1～P

T-nが、例えば外部記憶装置93上のリードオンリー（read only）のデータをファイルアクセスする場合には、上記のようにOSサービス手段50P1～50Pnによる処理を経ることなく、各プロセッサP1～Pnから外部記憶装置93に直接アクセスするようにしてもよい。

【0101】

ここで、第1プロセッサ側10のプロセッサP0上の逐次タスクSTからのファイルアクセス要求については、OSサービス手段によることなく直接シングルプロセッサ向けOS30によって処理がなされる。以下、図5を参照して説明する。

【0102】

(1) プロセッサP0上の逐次タスクSTは、シングルプロセッサ向けOS30に対してファイルアクセスを要求する。

【0103】

(2) これにより、シングルプロセッサ向けOS30は、要求に基づいて外部記憶装置93等に対するファイルアクセス（writeやread等）を行う。このファイルアクセス処理は、シングルプロセッサ向けOS30が本来備えているファイルアクセスサービスをそのまま利用して実行される。

【0104】

(3) ファイルアクセス処理が終了すると、シングルプロセッサ向けOS30は、逐次タスクSTに対してファイルアクセス要求に対する戻り値を返し、処理を戻す。

【0105】

逐次タスクSTからのOSサービス要求に対しては排他制御等の必要がないので、余計なオーバーヘッドが発生しない。

【0106】

制御処理中継手段60による処理動作について、図6及び図7を参照して説明する。

【0107】

まず、制御処理中継手段60の構成を図6に示す。図示のように、制御処理中

継手段 60 は、各プロセッサ P0 ~ Pn に対応した割り込み制御装置 61 P0 ~ 61 Pn と、各プロセッサ P0 ~ Pn に対応した通信領域 62 P0 ~ 62 Pn とを備えて構成される。ここで、通信領域 62 P0 ~ 62 Pn は、主記憶装置 92 内に確保される。

【0108】

さらに、割り込み装置 61 P0 ~ 61 Pn は、他のプロセッサに対して割り込みを指示する割込指示部 61 a、割り込みの指示により割り込みがあった旨の情報を保持する割込状態保持部 61 b、割り込みをクリアする割込取消部 61 c により構成される。

【0109】

また、通信領域 62 P0 ~ 62 Pn は、通信元のプロセッサからの通信理由情報を保持する通信理由保持領域 62 a、通信するための通信データを保持する通信データ保持領域 62 b、通信を確保するために通信領域に対してロックをかけるための排他制御領域 62 c により構成される。

【0110】

ここで、通信データ持領域 62 b には、通信する通信データ（要求内容を含む必要データ）が格納される主記憶装置 92 へのポインタが格納されることになる。

【0111】

第 1 プロセッサ側 10 の並列処理手段 40 P0 から第 2 プロセッサ側 20 のプロセッサ P1 への通信処理を例にとり、図 7 を参照してその動作を説明する。

【0112】

(1) 並列処理手段 40 P0 は、プロセッサ P1 用の通信領域 62 P1 の排他制御領域 62 c にロックをかける。具体的には、排他制御領域 62 c に格納される lock 変数をロック状態とすることにより、当該プロセッサ P1 用の通信領域 62 P1 が他のプロセッサで使用されるのを排する。

【0113】

他のプロセッサによって既にロックされている場合には、ロックが解除されるのを待つ。

【0114】

(2) 通信領域 62P1 のロックが取得できた場合、並列処理手段 40P0 は、通信理由情報と通信データ（要求内容を含む必要データ）を、通信領域 62P1 の通信理由保持領域 62a と通信データ保持領域 62b のそれぞれに格納する。

【0115】

格納される通信理由情報としては、上述したようにタスク生成のための通信処理であれば、「並列処理」を示す情報（例えば、並列処理に対応して予め定められている数値等のデータ）が格納される。

【0116】

(3) 並列処理手段 40P0 は、自身の割込制御装置 61P0 の割込指示部 61a に対して、プロセッサ P1 への割り込みを指示する。

【0117】

(4) 割込制御装置 61P0 の割込指示部 61a は、プロセッサ P1 に対応する割込制御装置 61P1 の割込状態保持部 61b に、割り込みを示す情報をセットする。これにより、プロセッサ P1 が割り込み状態となる。

【0118】

(5) プロセッサ P1 の並列処理手段 40P1 は、制御処理中継手段 60 の割込制御装置 61P1 の割込状態保持部 61b の状態より、割り込みがセットされたことを判別する。

【0119】

(6) プロセッサ P1 の並列処理手段 40P1 は、制御処理中継手段 60 の自身の割込状態保持部 61b の割り込み情報をクリアすることにより、自身の割り込み状態を解除する。

【0120】

(7) プロセッサ P1 の並列処理手段 40P1 は、さらに、制御処理中継手段 60 の自身の通信領域 62P1 の通信理由保持領域 62a と通信データ保持領域 62b から、通信理由情報と通信データ（要求内容を含む必要データ）をそれぞれ取得する。

【0121】

(8) プロセッサ P1 の並列処理手段 40 P1 は、次の通信を受理できるようになった後、自身の通信領域 62 P1 の排他制御領域 62 c のロックを解除する。具体的には、排他制御領域 62 c に格納される lock 変数をアンロック状態とすることにより、当該プロセッサ P1 用の通信領域 62 P1 が他のプロセッサから使用できるように解放する。

【0122】

以上のように、制御処理中継手段 60 を用いることにより、第1プロセッサ側 10 と第2プロセッサ側 20 間で、制御信号やデータのやり取りが実現される。

【0123】

また、代理手段 70 によるシグナル通知処理動作について、図8を参照して説明する。

【0124】

代理手段 70 は、シングルプロセッサ向け OS 30 が第2プロセッサ側 20 に生成したタスク PT-1 ~ PT-n とシグナル（制御信号）による通信を行うための機能を有する。この代理手段 70 には、対応するタスク PT-1 ~ PT-n のタスク番号が保持されている。

【0125】

なお、複数のタスク PT-1 ~ PT-n に一対一に代理手段 70 を関連付けてもよいし、一つの代理手段 70 に複数のタスク PT-1 ~ PT-n を関連付けてもよい。

【0126】

(1) 第1プロセッサ側 10 の逐次タスク ST が、シングルプロセッサ向け OS 30 に対してシグナル通知のサービスを要求する。このシグナル通知によって通信されるシグナルは、シングルプロセッサ向け OS 30 が本来有するサービス機能により、第2プロセッサ側 20 の各プロセッサ P1 ~ Pn 上で起動されているタスク PT-1 ~ PT-n を制御するための制御信号である。

【0127】

(2) シングルプロセッサ向け OS 30 は、シグナルを対応する代理手段 70

に通知する。

【0128】

(3) 代理手段70は、通知されたシグナルを制御処理中継手段60を通じて、自身に対応する第2プロセッサ側20のタスクPT-1の存在するプロセッサP1～Pnへ通知する。ここで、通信理由情報としては、「並列処理」が設定される。

【0129】

(4) 制御処理中継手段60への通信理由情報が「並列処理」であることから、第2プロセッサ側20の対応するプロセッサの並列処理手段40P1～40Pnは、制御処理中継手段60からシグナル通知要求を取得する。

【0130】

(5) シグナル通知要求を取得した並列処理手段40P1～40Pnは、対応するタスクPT-1～PTnに対してシグナル（制御信号）を通知する。

【0131】

以上により、代理手段70が第2プロセッサ側20のプロセッサP0～Pn上のタスクPT-1～PTnとタスク番号によって関連付けられているので、第1プロセッサ側10ではタスクPT-1～PTnに対してシングルプロセッサ向けOS30によるシグナル通知のサービスを実行することが可能となる。

【0132】

最後に、第2プロセッサ側20におけるタスク相互の連携動作について、図9と図10を参照して説明する。

【0133】

タスク間の連携を取るための並列処理手段40P1～40Pnの構成を図9に示す。図示のように、並列処理手段40P1～40Pnは、相互排他手段41P1～41Pnを備えている。

【0134】

また、タスクの連携を取るため、主記憶装置92内には、排他制御領域43とタスク管理内容保持領域44を有する。

【0135】

(1) プロセッサ P 1 上のタスク P T - 1 は、自身の並列処理手段 4 0 P 1 に対して所望の並列処理を要求する。

【0136】

(2) プロセッサ P 1 の並列処理手段 4 0 P 1 は、相互排他手段 4 1 P 1 に並列処理に用いる資源の確保のためにロック取得を要求する。

【0137】

(3) 相互排他手段 4 1 P 1 は、主記憶装置 9 2 の排他制御領域 4 3 を用いてロックを取得する。

【0138】

(4) 相互排他手段 4 1 P 1 は、並列処理手段 4 0 P 1 に対してロック取得ができた旨を通知する。

【0139】

(5) 並列処理手段 4 0 P 1 は、タスク管理内容保持領域 4 4 にプロセッサ P 1 上の並列処理を要求したタスク P T - 1 の管理内容（要求内容、タスク番号その他の必要なデータ）を保持する。

【0140】

(6) 並列処理手段 4 0 P 1 は、優先度の低いタスクを実行している他のプロセッサ P 2 ~ P n へ割り込みをかける。ここでは、プロセッサ P n への割り込みがかけられたものとする。

【0141】

(7) 並列処理手段 4 0 P 1 は、相互排他手段 4 1 P 1 に対して上記確保したロックの解除を要求する。

【0142】

(8) 相互排他手段 4 1 P 1 は、主記憶装置 9 2 の排他制御領域 4 3 のロックを解除する。

【0143】

(9) 相互排他手段 4 1 P 1 は、並列処理手段 4 0 P 1 に対してロックの解除完了を通知する。

【0144】

(10) 並列処理手段 40P1 は、タスク PT-1 に処理を戻す。

【0145】

(11) プロセッサ Pn の並列処理手段 40Pn は、相互排他手段 41Pn に対して並列処理に用いる資源の確保のためのロック取得を要求する。

【0146】

(12) 相互排他手段 41Pn は、主記憶装置 92 の排他制御領域 43 を用いてロックを取得する。

【0147】

(13) 相互排他手段 41Pn は、並列処理手段 40Pn に対してロックの取得完了を通知する。

【0148】

(14) 並列処理手段 40Pn は、タスク管理内容保持領域 44 内に格納されているタスク管理内容を取得する。

【0149】

(15) 並列処理手段 40Pn は、タスク管理内容に基づいて新たなタスクを生成起動する。

【0150】

(16) 並列処理手段 40Pn は、相互排他手段 41Pn に対して上記確保したロックの解除を要求する。

【0151】

(17) 相互排他手段 41Pn は、主記憶装置 92 の排他制御領域 43 のロックを解除する。

【0152】

(18) 相互排他手段 41Pn は、並列処理手段 40Pn に対してロックの解除完了を通知する。

【0153】

以上のタスク連携動作により、あるプロセッサ P1～Pn で起動しているタスク PT-1～PT-n が、自身の処理単位を他のプロセッサに対して並列処理させることが可能となる。

【0154】

上記のように構成される並列処理システムにおける適用例としては、例えばマルチプロセッサの携帯端末において、アプリケーションを動作させる場合を説明する。

【0155】

ここで、アプリケーションが携帯端末上で映像（動画）と音声と同時に再生する機能を有し、そのうち音声出力処理を逐次タスクAとして、映像出力処理を並列化タスクBとして第1プロセッサ側10のプロセッサP0上で動作させる場合を想定して、図11を参照して説明する。

【0156】

並列化タスクBは、上述した並列処理手段によって、第2プロセッサ側20のプロセッサP1～Pnの何れかにタスクCとして生成され起動される。

【0157】

生成されたタスクCは、OSサービス手段のファイルアクセス処理によって映像の画面出力を実行する。

【0158】

また、逐次タスクAによる音声出力は、シングルタスクOS30のサービスによって音声の出力が実行される。

【0159】

音声と映像の同期を定期的にする場合には、上述した代理手段70によるシグナル通知動作によって逐次タスクAとタスクC間のシグナル通知を行うことで同期をとる。

【0160】

また、タスクCにおいて、映像に編集を加える処理を実行する場合には、その編集処理をタスク連携処理によって他のプロセッサ上にタスクDとして生成起動する。

【0161】

タスクCによる映像出力処理が終了した場合には、タスクCからファイルアクセス動作によってシングルプロセッサ向けOSに終了通知を行う。

【0162】

本実施の形態においては、並列処理手段40P0～40Pn、OSサービス手段50P0～50Pn、制御処理中継手段60及び代理手段70については、モジュールとして実装することにより、シングルプロセッサ向けOS30及びプロセッサP0上で実行されるアプリケーションに改造を加えることなく、マルチプロセッサのシステム構成上で、シングルプロセッサ向けOS30及びアプリケーションをオーバーヘッドなく動作させることができると共に、マルチプロセッサによる並列処理の恩恵を受けることが可能となる。

【0163】

次に、本発明の第2の実施の形態による並列処理システムについて、図12以下を参照して説明する。図12は、この第2の実施の形態による並列処理システムの構成を示すブロック図であり、図1と共通の構成要素については同じ符号を用いている。図12に示す第2の実施の形態では、第1の実施の形態における並列処理手段40P0～40Pnを、それぞれ並列処理通信手段400P0～400Pnと制御代理手段500P0～500Pnとで構成した場合を示している。

【0164】

図12に示すように、本実施の形態による並列処理システムにおいて、システムバス91で接続される複数のプロセッサ(CPU)P0～Pn(nは1以上の整数)からなるマルチプロセッサを、論理的に第1プロセッサ側100と第2プロセッサ側200の二組に分けて構成している点については、第1の実施の形態と同様である。

【0165】

この第2の実施の形態においては、第1プロセッサ側100のプロセッサP0及び第2プロセッサ200のプロセッサ(CPU)P0～Pnの各プロセッサ上で動作するシングルプロセッサ向けOS300P0～300Pnを実装している。

【0166】

また、第1プロセッサ側100のプロセッサP0及び第2プロセッサ側200のプロセッサP1～Pnのそれぞれに、並列処理を行うための並列処理通信手段

400P0～400Pnと制御代理手段500P0～500Pnを実装して構成される。また、システムバス91には、各プロセッサP0～Pnが共有する共有メモリとしての主記憶装置92及びディスク装置等の外部記憶装置93が接続されている。

【0167】

また、第1プロセッサ側100と第2プロセッサ側200間で、制御信号やデータのやり取りを行うための制御処理中継手段600が独自に設けられている。

【0168】

なお、この実施の形態では、第1の実施の形態で示した第1プロセッサ側100のシングルプロセッサ向けOS300P0が第2プロセッサ側200で実行されるプロセスと通信するための代理手段70については、第1の実施の形態と全く同じ機能を実行するため、便宜上省略している。

【0169】

また、第1の実施の形態では、第2プロセッサ側20のプロセッサP1～Pn上のタスクからのファイルアクセス要求に対してタスクのswitch（スイッチ）がなされることを説明したが、本実施の形態においても、第2プロセッサ側200からのファイルアクセスが可能であり、ファイルアクセス要求を行なった第2プロセッサ側200のプロセスPP-1～PPnは、第2プロセッサ側200のシングルプロセッサ向けOS300P1～300Pnによってプロセスのswitch（スイッチ）が行なわれる。

【0170】

上記各プロセッサP0～Pnに実装されるシングルプロセッサ向けOS300P0～300Pnについては、リアルタイムOSのようなメモリ保護機能を実現する仮想記憶機構を持たないOSだけではなく、既存のOSであるLinuxやWindows(R)等といった仮想記憶機構を備えるOSも用いることが可能であり、各プロセッサP0～Pnの全て、もしくは一部においてメモリ保護機構の実現が可能である点で第1の実施の形態と相違する。

【0171】

なお、シングルプロセッサ向けOS300P0～300Pnは、全部が同じ種

類のOSである必要はなく、互いに異なる種類のOSであってもよい。

【0172】

この実施の形態では、並列処理を行うための処理単位であるタスクが互いのプロセッサ間でメモリ保護されており、その意味で第1の実施の形態のタスクと区別するためにプロセスと称することとする。

【0173】

本実施の形態による並列処理システムにおいては、第1プロセッサ側100のシングルプロセッサ向けOS300P0上で動作するアプリケーションの処理単位のうち、並列化を行えないプロセス（逐次プロセスSP）を第1プロセッサ側100のプロセッサP0で処理し、アプリケーション内で並列化を行えるタスクを第2プロセッサ側200において新たなプロセスPP-1～PP-nとして生成し並列処理をさせる。

【0174】

並列処理通信手段400P0及び並列処理通信手段400P1～400Pnは、プロセスとシングルプロセッサ向けOSOS300P0～300Pnとの間で、プロセスの生成、起動、停止、終了、削除、及びその他のプロセスに関する制御情報の受け渡しを行う機能を有する。

【0175】

ここで、第1プロセッサ側100と第2プロセッサ側200間においては、制御処理中継手段600を介してタスクの生成、起動、停止、終了、削除等に関する制御情報やデータをやり取りする。

【0176】

また、制御代理手段500P0及び制御代理手段500P1～500Pnは、シングルプロセッサ向けOSOS300P0～300Pnからプロセスへの処理要求を入手してプロセスを起動する機能を有する。

【0177】

制御処理中継手段600は、第1プロセッサ側10と第2プロセッサ側20間で、制御信号やデータのやり取りを行うための手段であり、複数のプロセッサで並列処理される複数のプロセス間の制御に使用される。

【0178】

以下、上記のように構成される第2の実施の形態による並列処理システムの動作について、図面を参照して詳しく説明する。

【0179】

ここでは、第1プロセッサ側100のシングルプロセッサ向けOS300P0上でアプリケーションが動作し、そのアプリケーションの処理単位のうち、第1プロセッサ側100のプロセッサP0で処理する処理単位を逐次プロセスSPとし、アプリケーション内で並列化を行えるプロセスであって、第2プロセッサ側200に対してタスクPP-1～PP-nとして並列処理をさせる処理単位を並列化プロセスPPとする。

【0180】

この第2の実施の形態においては、あるプロセス（タスク）を起動した後に、呼び出し側のプロセス（タスク）である並列化プロセスPPが、起動したプロセスPP-1～PP-nの終了を待ち合わせる並列処理の同期起動と、呼び出し側のプロセス（タスク）である並列化プロセスPPが、起動したプロセスPP-1～PP-nの終了を待ち合わせなくとよい並列処理の非同期起動とが可能である。

【0181】

まず、プロセスの並列処理の同期起動の動作について、図13を参照して説明する。

【0182】

ここでは、第1プロセッサ側100の並列化プロセスPPが、第2プロセッサ側200のプロセッサPk（ $1 \leq k \leq n$ ）において、第2プロセッサ側200で起動させる処理単位であるプロセスPP-kとして予め生成されているものとする。

【0183】

(1) 第2プロセッサ側200のプロセッサPk（ $1 \leq k \leq n$ ）上のプロセスPP-kが、並列処理通信手段400Pkに対して、自プロセスに属する処理の情報を登録する。これにより、プロセスPP-kは待機状態となっている。

【0184】

(2) 並列処理通信手段400Pkは、シングルプロセッサ向けOS300Pk経由で主記憶装置92（共有メモリ）上に必要なデータをアクセスする要求を行う。

【0185】

(3) これにより、プロセスP P-kに属する処理の情報が主記憶装置92（共有メモリ）に格納される。

【0186】

(4) プロセッサP0上のプロセスP Pは、並列処理通信手段400P0に対して同期的な処理を依頼する。その後、プロセスP Pは待機状態となる。

【0187】

(5) 依頼を受けた並列処理通信手段400P0は、シングルプロセッサ向けOS300P0経由で主記憶装置92（共有メモリ）上のデータに対するアクセス要求を行う。

【0188】

(6) シングルプロセッサ向けOS300P0により主記憶装置92（共有メモリ）がアクセスされる。

【0189】

(7) 並列処理通信手段400P0は、主記憶装置92（共有メモリ）からプロセスP P-kに属する処理の情報を取得する。これにより、プロセスP P-kを実行するプロセッサ番号とプロセス番号が取得される。

【0190】

(8) 並列処理通信手段400P0は、シングルプロセッサ向けOS300P0経由で主記憶装置92（共有メモリ）から取得した情報に基づいて通信依頼を要求する。

【0191】

(9) シングルプロセッサ向けOS300P0により、通信に必要な情報が制御処理中継手段600に設定される。

【0192】

(10) これにより、制御処理中継手段 600 からプロセッサ P k のシングルプロセッサ向け OS 300 P k に対して処理要求が送信される。

【0193】

(11) プロセッサ P k の制御代理手段 500 P k は、シングルプロセッサ向け OS 300 P k 経由で送信された処理要求を入手する。

【0194】

(12) そして、制御代理手段 500 P k は、入手した処理要求によりプロセス P P - k を起動する。

【0195】

(13) これにより、待機状態にあったプロセス P P - k は処理を実行する。

【0196】

(14) 処理が終了後、プロセス P P - k は、その旨の情報を並列処理通信手段 400 P k に通知する。

【0197】

(15) 通知を受けた並列処理通信手段 400 P k は、シングルプロセッサ向け OS 300 P k 経由で通知された情報に基づいて通信依頼を要求する。

【0198】

(16) シングルプロセッサ向け OS 300 P k により、通信に必要な情報が制御処理中継手段 600 に設定される。

【0199】

(17) これにより、制御処理中継手段 600 からプロセッサ P 0 のシングルプロセッサ向け OS 300 P 0 に対して処理要求が送信される。

【0200】

(18) プロセッサ P 0 の制御代理手段 500 P 0 は、シングルプロセッサ向け OS 300 P 0 経由で送信された処理要求を入手する。

【0201】

(19) そして、制御代理手段 500 P 0 は、入手した処理要求によりプロセス P P を起動する。

【0202】

以上により、第1プロセッサ側100のプロセッサP0上のプロセスPPと第2プロセッサ側200のプロセッサPP-kとの間での同期的な処理が実現される。

【0203】

次に、プロセスの並列処理の非同期起動の動作について、図14を参照して説明する。

【0204】

ここでも、第1プロセッサ側100の並列化プロセスPPが、第2プロセッサ側200のプロセッサPk ($1 \leq k \leq n$) において、第2プロセッサ側200で起動させる処理単位であるプロセスPP-kとして予め生成されているものとする。

【0205】

(1) 処理を行っていた第2プロセッサ側200のプロセッサPk ($1 \leq k \leq n$) 上のプロセスPP-kが、並列処理通信手段400Pkに対して、自プロセスに属する処理の情報を登録する。これにより、プロセスPP-kは待機状態となる。

【0206】

(2) 並列処理通信手段400Pkは、シングルプロセッサ向けOS300Pk経由で主記憶装置92（共有メモリ）上に必要なデータをアクセスする要求を行う。

【0207】

(3) これにより、プロセスPP-kに属する処理の情報が主記憶装置92（共有メモリ）に格納される。

【0208】

(4) プロセッサP0上のプロセスPPは、並列処理通信手段400P0に対して非同期的な処理を依頼する。この場合、プロセスPPは待機状態となることなくその後も処理を継続する。

【0209】

(5) 依頼を受けた並列処理通信手段400P0は、シングルプロセッサ向け

OS300P0 経由で主記憶装置 92（共有メモリ）上のデータに対するアクセス要求を行う。

【0210】

（6）シングルプロセッサ向け OS300P0 により主記憶装置 92（共有メモリ）がアクセスされる。

【0211】

（7）並列処理通信手段 400P0 は、主記憶装置 92（共有メモリ）からプロセス P P - k に属する処理の情報を取得する。これにより、プロセス P P - k を実行するプロセッサ番号とプロセス番号が取得される。

【0212】

（8）並列処理通信手段 400P0 は、シングルプロセッサ向け OS300P0 経由で主記憶装置 92（共有メモリ）から取得した情報に基づいて通信依頼を要求する。

【0213】

（9）シングルプロセッサ向け OS300P0 により、通信に必要な情報が制御処理中継手段 600 に設定される。

【0214】

（10）これにより、制御処理中継手段 600 からプロセッサ P k のシングルプロセッサ向け OS300P k に対して処理要求が送信される。

【0215】

（11）プロセッサ P k の制御代理手段 500P k は、シングルプロセッサ向け OS300P k 経由で送信された処理要求を入手する。

【0216】

（12）そして、制御代理手段 500P k は、入手した処理要求によりプロセス P P - k を起動する。

【0217】

（13）これにより、待機状態にあったプロセス P P - k は処理を実行する。

【0218】

以上により、第 1 プロセッサ側 100 のプロセッサ P 0 上のプロセス P P と第

2 プロセッサ側 200 のプロセッサ P P - k との間での非同期的な処理が実現される。

【0219】

さらに、第2の実施の形態による並列処理システムにおける制御処理中継手段 600 による処理動作について、図15及び図16を参照して説明する。

【0220】

まず、制御処理中継手段 600 の構成を図15に示す。図示のように、制御処理中継手段 600 は、各プロセッサ P 0 ~ P n に対応した割り込み制御装置 601 P 0 ~ 601 P n と、各プロセッサ P 0 ~ P n に対応した通信領域 602 P 0 ~ 602 P n とを備えて構成される。ここで、通信領域 602 P 0 ~ 602 P n は、主記憶装置 92 内に確保される。

【0221】

上記割り込み制御装置 601 P 0 ~ 601 P n については、図6に示した第1の実施の形態の制御処理中継手段 60 の構成と同様であり、割込指示部 601 a 、割込状態保持部 601 b 、割込取消部 601 c により構成される。

【0222】

また、通信領域 602 P 0 ~ 602 P n についても、図6に示した第1の実施の形態と基本的に同じ構成であり、図6の通信理由保持領域 62 a と通信データ保持領域 62 b とを組み合わせた通信キュー 602 a と、排他制御領域 602 c により構成される。

【0223】

第1プロセッサ側 100 の並列処理通信手段 400 P 0 から第2プロセッサ側 200 のプロセッサ P n への通信処理を例にとり、図16を参照してその動作を説明する。

【0224】

(1) 並列処理通信手段 400 P 0 は、プロセッサ P 1 用の通信領域 602 P n の排他制御領域 602 b にロックをかける。具体的には、排他制御領域 602 b に格納される lock 変数をロック状態とすることにより、当該プロセッサ P n 用の通信領域 602 P n が他のプロセッサで使用されるのを排する。

【0225】

(2) 並列処理通信手段 400P0は、通信理由情報と通信データ（要求内容を含む必要データ）を、通信領域 602Pnの通信キュー 602aに格納する。

【0226】

格納される通信理由情報としては、上述したようにプロセスの生成起動のための通信処理であれば、「プロセス起動」を示す情報（例えば、並列処理に対応して予め定められている数値等のデータ）が格納される。

【0227】

(3) (1)の時点で通信領域 602Pnの通信キュー 602aが空であった場合、並列処理通信手段 400P0は、自身の割込制御装置 601P0の割込指示部 601aに対して、プロセッサ Pnへの割り込みを指示する。

【0228】

(4) プロセッサ P0の並列処理通信手段 400P0は、プロセッサ Pn用の通信領域 602Pnの排他制御領域 602bをアンロック状態とする。(1)の時点で通信領域 602Pnの通信キュー 602aが空でない場合には、割り込みの必要がないので、ここで処理を終了する。

【0229】

(5) 割込制御装置 601P0の割込指示部 601aは、プロセッサ Pnに対応する割込制御装置 601Pnの割込状態保持部 601bに、割り込みを示す情報をセットする。これにより、プロセッサ Pnが割り込み状態となる。

【0230】

(6) プロセッサ Pnのシングルプロセッサ向け OS 300pnは、プロセッサ Pn用の割込制御装置 601Pnの割込状態保持部 601bの状態より、割り込みを受理する。

【0231】

(7) プロセッサ Pnのシングルプロセッサ向け OS 300pnは、割込制御装置 601Pnの割込取消部 601cによって自身の割込状態保持部 601bの割り込み情報をクリアすることにより、自身の割り込み状態を解除する。

【0232】

(8) プロセッサ P_nのシングルプロセッサ向け OS 300 p_nは、自身の制御代理手段 500 P_nを起動する。

【0233】

(9) プロセッサ P_nの制御代理手段 500 P_nは、自身の通信領域 602 P_nの排他制御領域 602 b にロックをセットする。

【0234】

(10) プロセッサ P_nの制御代理手段 500 P_nは、さらに、自身の通信領域 602 P_nの通信キュー 602 a から、通信理由情報と通信データをそれぞれ削除する。

【0235】

(11) プロセッサ P_nの制御代理手段 500 P_nが、プロセッサ P_n用の通信領域 602 P_nの排他制御領域 602 b をアンロックとする。これにより、当該プロセッサ P_n用の通信領域 602 P_nが他のプロセッサから使用できるようになる。

【0236】

以上のように、制御処理中継手段 600 を用いることにより、第1プロセッサ側 100 と第2プロセッサ側 200 間で、制御信号やデータのやり取りが実現される。

【0237】

次に、本発明の第3の実施の形態による並列処理システムについて、図17以下を参照して説明する。図17は、この第3の実施の形態による並列処理システムの構成を示すブロック図であり、図13と共通の構成要素については同じ符号を用いている。

【0238】

上記説明した第2の実施の形態における同期的な処理では、プロセス相互において、一方のプロセスが他方のプロセスによって主記憶装置 92 上のデータが更新されたかどうかのチェックを繰り返す必要があり、その繰り返しの分だけ余計な処理が発生していたが、本実施の形態では、そのような余計な処理を必要としない高性能なプロセス間の同期とデータ受け渡しを可能にするものである。

【0239】

図17に示すように、第3の実施の形態による並列処理システムにおいては、第2の実施の形態と同様に、システムバス91で接続される複数のプロセッサ（CPU）P0～Pn（nは1以上の整数）からなるマルチプロセッサを、論理的に第1プロセッサ側100と第2プロセッサ側200の二組に分けて構成し、第1プロセッサ側100のプロセッサP0及び第2プロセッサ200のプロセッサ（CPU）P0～Pnの各プロセッサ上で動作するシングルプロセッサ向けOS 300P0～300Pnを実装している。

【0240】

この第3の実施の形態では、第2の実施の形態において、第1プロセッサ側100のプロセッサP0及び第2プロセッサ側200のプロセッサP1～Pnの並列処理を行うための並列処理通信手段400P0～400Pnと制御代理手段500P0～500Pnに加えて、第1プロセッサ側100のプロセッサP0及び第2プロセッサ側200のプロセッサP1～Pn上で実行される各プロセス間での通信を実現するためのプロセス間通信手段1000P0～1000Pnを新たに備えた点を特徴としている。

【0241】

すなわち、本実施の形態では、これまでマルチプロセッサ向けOSに実装されていたプロセス間通信の機能を、マルチプロセッサ上でシングルプロセッサ向けOS及びアプリケーションを動作させるシングルプロセッサ向けOSによる並列処理システム上で実現するものであり、ユーザレベルによるプロセス間通信機能の実装を可能にしている。

【0242】

なお、この実施の形態でも、第1の実施の形態で示した第1プロセッサ側100のシングルプロセッサ向けOS 300P0が第2プロセッサ側200で実行されるプロセスと通信するための代理手段70については、第1の実施の形態と全く同じ機能を実行するため、便宜上省略している。

【0243】

上記プロセス間通信手段1000P0～1000Pn以外の構成要素に関する

構成及び動作については、上述した第2の実施の形態と全く同じであるので、以下では、プロセス間通信手段1000P0～1000Pnについてのみ説明することとする。

【0244】

上記プロセス間通信手段1000P0～1000Pnは、セマフォ（semaphore）やメッセージキュー等の方式を用いることにより、プロセッサP1～Pn上で実行される各プロセス間での通信を実現するものである。

【0245】

ここでは、プロセス間通信手段1000P0～1000Pnがセマフォ方式を用いてプロセス間通信を行なう場合について説明する。

【0246】

図18に示すように、第2プロセッサ側200のプロセッサPmとプロセッサPn上でそれぞれプロセスPP-mとプロセスPP-nが実行されている場合において、プロセスPP-mとプロセスPP-nがプロセス間通信手段1000Pmと1000Pnによるセマフォ方式を用いて通信を行う場合について説明する。

【0247】

セマフォ方式とは、複数のプロセスが同時実行されるマルチタスクOSにおいて、複数プロセス間で通信して同期をとるための方式であり、セマフォは一種の共有フラグ（カウンタ）で、同期をとるプロセス同士がこのカウンタに注目し、カウンタの変化に応じて処理を行なうようにすることで通信（同期）を実現する。

【0248】

まず、図18において、プロセッサPm上のプロセスPP-mが主記憶装置92のセマフォ情報領域（共有フラグ）をup又はdownする場合の動作を説明する。ここで、セマフォupは、セマフォ情報領域のカウンタ値をインクリメントし、セマフォdownは、セマフォ情報領域のカウンタ値をデクリメントする。セマフォdownできない場合、当該プロセスはスリープして待ち状態となり、セマフォupによってウェイクアップされる。

【0249】

(1) プロセッサ P_m上のプロセス P P -mがセマフォ (セマフォカウンタ) を u p 又は d o w n する場合、プロセス間通信手段 1 0 0 0 P_mにその旨を要求する。

【0250】

(2) 要求を受けたプロセス間通信手段 1 0 0 0 P_mは、主記憶装置 9 2 に確保されているセマフォ番号で区別される共有フラグとしてのセマフォ情報領域 # N (このセマフォ情報領域は、システムによって予め決定される数だけ確保される) にアクセスすることにより、セマフォを u p 又は d o w n できることが判明する。

【0251】

次に、図 1 9 において、プロセッサ P_m上のプロセス P P -mがセマフォ (セマフォフラグ) を d o w n する場合の動作を説明する。

【0252】

(1) プロセッサ P_m上のプロセス P P -mがセマフォ (セマフォフラグ) を d o w n しようとプロセス間通信手段 1 0 0 0 P_mに要求する。

【0253】

(2) 要求を受けたプロセス間通信手段 1 0 0 0 P_mは、主記憶装置 9 2 に確保されているセマフォ情報領域 # N にアクセスする。ここでは、セマフォ情報領域 # N には、セマフォのカウンタ値として初期値「0」が設定されており、d o w n しようとしても d o w n できないように予め定められている。よって、セマフォを d o w n できないことが判明する。

【0254】

(3) プロセス間通信手段 1 0 0 0 P_mは、シングルプロセッサ向け O S 3 0 0 P_mに対しシステムコールを用いて上記プロセス P P -mをスリープさせるように要求する。

【0255】

(4) シングルプロセッサ向け O S 3 0 0 P_mはシステムコールを用いてプロセス P P -mをスリープさせ、ランキューから外す。

【0256】

すなわち、上記の場合、プロセス P P - m はセマフォを d o w n できないのでスリープする。

【0257】

さらに、図 20 において、上記のようにプロセス P P - m がスリープした状態で、同じプロセッサ P m 上のプロセス P P - m 1 がセマフォを u p する場合の動作を説明する。

【0258】

(1) プロセス P P - m 1 がセマフォの u p をプロセス間通信手段 1000 P m に要求する。

【0259】

(2) プロセス間通信手段 1000 P m は、主記憶装置 92 に確保されているセマフォ情報領域 # N にアクセスすることにより、セマフォ待ちのプロセス P P - m が存在することが判明する。

【0260】

(3) さらに、プロセス間通信手段 1000 P m は、制御処理中継手段 600 に対してメッセージ（制御メッセージ）を送信することで、プロセス P P - m のウェイクアップを要求する。

【0261】

(4) 制御代理手段 500 P m が、制御処理中継手段 600 からメッセージを受信する。この制御処理中継手段 600 によるメッセージの中継は、図 15、図 16 で説明した動作に基づいて行われる。

【0262】

(5) 制御代理手段 500 P m は、メッセージからセマフォ待ちのプロセスのウェイクアップ要求であると判断し、シングルプロセッサ向け OS 300 P m に対しシステムコールを用いて待ち状態の上記プロセス P P - m をウェイクアップさせるよう要求する。

【0263】

(6) シングルプロセッサ向け OS 300 P m は、システムコールを用いて要

求されたプロセス P P - m をウェイクアップし、ランキューへつなぐ。この場合、ウェイクアップされたプロセス P P - m は再度セマフォを d o w n しようとする。

【0264】

なお、上記処理 (3)、(4) における制御処理中継手段 600 による制御メッセージの中継を利用することなく、シングルプロセッサ向け O S 300 P m の通信機能を用いて、プロセス間通信手段 1000 P m から制御代理手段 500 P m に対して直接プロセス P P - m のウェイクアップを要求するようにすれば、処理 (3)、(4) が簡略されて高速化が可能である。

【0265】

図 21 において、上記のようにプロセッサ P m 上のプロセス P P - m がスリープした状態で、他のプロセッサ P n 上のプロセス P P - n がセマフォを u p する場合の動作を説明する。

【0266】

(1) プロセッサ P n 上のプロセス P P - n がセマフォの u p を自プロセッサ P n のプロセス間通信手段 1000 P n に要求する。

【0267】

(2) プロセス間通信手段 1000 P n は、主記憶装置 92 に確保されているセマフォ情報領域 # N にアクセスすることにより、セマフォ待ちのプロセス P P - m が存在することが判明する。

【0268】

(3) さらに、プロセス間通信手段 1000 P n は、制御処理中継手段 600 に対してメッセージ (制御メッセージ) を送信することで、プロセス P P - m のウェイクアップを要求する。

【0269】

(4) 次に、プロセッサ P m の制御代理手段 500 P m が、制御処理中継手段 600 からメッセージを受信する。この制御処理中継手段 600 によるメッセージの中継は、図 15、図 16 で説明した動作に基づいて行われる。

【0270】

(5) 制御代理手段 500 Pm は、メッセージからセマフォ待ちのプロセスのウェイクアップ要求であると判断し、シングルプロセッサ向け OS 300 Pm に対しシステムコールを用いて待ち状態の上記プロセス PP-m をウェイクアップさせるよう要求する。

【0271】

(6) シングルプロセッサ向け OS 300 Pm は、システムコールを用いて要求されたプロセス PP-m をウェイクアップし、ランキューへつなぐ。この場合、ウェイクアップされたプロセス PP-m は再度セマフォを down しようと試みる。

【0272】

次に、図 22 に示すように、第 2 プロセッサ側 200 のプロセッサ Pm とプロセッサ Pn 上でそれぞれプロセス PP-m とプロセス PP-n が実行されている場合において、プロセス PP-m とプロセス PP-n がプロセス間通信手段 1000 Pm と 1000 Pn によるメッセージキュー方式を用いて通信を行う場合について説明する。

【0273】

メッセージキュー方式とは、複数のプロセス間で通信を行なう方法であって、名前の通り「キュー」を作成して、そこに情報としてのメッセージ（プロセス相互でやり取りされる処理データ）を格納していく方式である。メッセージキュー方式では、受け取り側のプロセスはそのメッセージを任意の順番で受け取ることができる。また、メッセージを何れかのプロセスが受け取るとそのメッセージはキューから消える。

【0274】

図 22 において、プロセッサ Pm 上のプロセス PP-m がメッセージキュー方式によってメッセージを受信する場合の動作を説明する。ここでは、取得すべきメッセージが主記憶装置 92 に確保されているキュー番号で区別されるキュー情報領域 #N（このキュー情報領域は、システムによって予め決定される数だけ確保される）に存在する場合である。

【0275】

(1) プロセッサ P_m上のプロセス P P-mがメッセージの受信をするためにプロセス間通信手段 1000 P_mにその旨を要求する。

【0276】

(2) 要求を受けたプロセス間通信手段 1000 P_mは、主記憶装置 92に確保されているキュー情報領域 #Nをアクセスし、メッセージを受信する。

【0277】

(3) そして、プロセス間通信手段 1000 P_mは、受信したメッセージを要求元のプロセス P P-mにコピーする。この時点で当該メッセージは主記憶装置 92に確保されているキュー情報領域 #Nから消去される。

【0278】

次に、図 23において、プロセッサ P_m上のプロセス P P-mがメッセージキュー方式によってメッセージを受信する場合の第2の動作を説明する。ここでは、取得すべきメッセージが主記憶装置 92に確保されているキュー情報領域 #Nに存在しない場合である。

【0279】

(1) プロセッサ P_m上のプロセス P P-mがメッセージの受信をするためにプロセス間通信手段 1000 P_mにその旨を要求する。

【0280】

(2) 要求を受けたプロセス間通信手段 1000 P_mは、主記憶装置 92に確保されているキュー情報領域 #Nにアクセスする。この場合、取得すべきメッセージがに確保されているキュー情報領域 #Nに存在せず、メッセージを受信できないことが判明する。

【0281】

(3) プロセス間通信手段 1000 P_mは、シングルプロセッサ向け OS 300 P_mに対しシステムコールを用いて上記プロセス P P-mをスリープさせるように要求する。

【0282】

(4) シングルプロセッサ向け OS 300 P_mはシステムコールを用いてプロセス P P-mをスリープさせ、ランキューから外す。

【0283】

すなわち、上記の場合、プロセス P P - m はメッセージを受信できないのでスリープする。

【0284】

図 24 において、プロセッサ P m 上のプロセス P P - m がメッセージキュー方式によってメッセージを送信する場合の動作を説明する。ここでは、送信しようとするメッセージ待ちのプロセスが存在せず、主記憶装置 92 に確保されているキュー情報領域 # N へのメッセージの送信のみを行なう場合である。

【0285】

(1) プロセッサ P m 上のプロセス P P - m がメッセージの送信するためにプロセス間通信手段 1000 P m にその旨を要求する。

【0286】

(2) 要求を受けたプロセス間通信手段 1000 P m は、主記憶装置 92 に確保されているキュー情報領域 # N にアクセスし、プロセス P P - m からのメッセージを格納する。

【0287】

さらに、図 25 において、プロセッサ P m 上のプロセスがメッセージキュー方式によってメッセージを送信する場合の動作を説明する。ここでは、送信しようとするメッセージ待ちのプロセスが存在する場合である。すなわち、図 23 の例のように、プロセス P P - m がメッセージを受信できずにスリープしている状態でプロセス P P - m 1 からメッセージが送信される場合である。

【0288】

(1) プロセス P P - m 1 がメッセージの送信をプロセス間通信手段 1000 P m に要求する。

【0289】

(2) プロセス間通信手段 1000 P m は、主記憶装置 92 に確保されているキュー情報領域 # N にアクセスし、プロセス P P - m 1 からのメッセージを格納する。ここで、メッセージ待ちのプロセス P P - m が存在することが判明する。

【0290】

(3) さらに、プロセス間通信手段 1000 Pm は、制御処理中継手段 600 に対して制御メッセージを送信することで、プロセス P P - m のウェイクアップを要求する。

【0291】

(4) 制御代理手段 500 Pm が、制御処理中継手段 600 から制御メッセージを受信する。

【0292】

(5) 制御代理手段 500 Pm は、制御メッセージからメッセージ待ちのプロセスのウェイクアップ要求であると判断し、シングルプロセッサ向け OS 300 Pm に対しシステムコールを用いて待ち状態の上記プロセス P P - m をウェイクアップさせるよう要求する。

【0293】

(6) シングルプロセッサ向け OS 300 Pm は、システムコールを用いて要求されたプロセス P P - m をウェイクアップし、ランキューへつなぐ。

【0294】

この場合、ウェイクアップされたプロセス P P - m は再度メッセージの受信を試みる。これにより、プロセス P P - m 1 からのメッセージがプロセス P P - m で受信される。

【0295】

なお、上記処理 (3)、(4) における制御処理中継手段 600 による制御メッセージの中継を利用することなく、シングルプロセッサ向け OS 300 Pm の通信機能を用いて、プロセス間通信手段 1000 Pm から制御代理手段 500 Pm に対して直接プロセス P P - m のウェイクアップを要求するようになれば、処理 (3)、(4) が簡略されて高速化が可能である。

【0296】

図 26 において、図 23 に示すようにプロセッサ Pm 上のプロセス P P - m がスリープしてメッセージ待ちの状態、他のプロセッサ Pn 上のプロセス P P - n がメッセージを送信する場合の動作を説明する。

【0297】

(1) プロセッサ P_n 上のプロセス $PP-n$ がメッセージの送信を自プロセッサ P_n のプロセス間通信手段 1000 P_n に要求する。

【0298】

(2) プロセス間通信手段 1000 P_n は、主記憶装置 92 に確保されているキュー情報領域 # N にアクセスし、プロセス $PP-n$ からのメッセージを格納する。ここで、メッセージ待ちのプロセス $PP-m$ が存在することが判明する。

【0299】

(3) さらに、プロセス間通信手段 1000 P_n は、制御処理中継手段 600 に対して制御メッセージを送信することで、プロセス $PP-m$ のウェイクアップを要求する。

【0300】

(4) 次いで、プロセッサ P_m の制御代理手段 500 P_m が、制御処理中継手段 600 から制御メッセージを受信する。

【0301】

(5) 制御代理手段 500 P_m は、制御メッセージからメッセージ待ちのプロセスのウェイクアップ要求であると判断し、シングルプロセッサ向け OS 300 P_m に対しシステムコールを用いて待ち状態の上記プロセス $PP-m$ をウェイクアップさせるよう要求する。

【0302】

(6) シングルプロセッサ向け OS 300 P_m は、システムコールを用いて要求されたプロセス $PP-m$ をウェイクアップし、ランキューへつなぐ。

【0303】

この場合、ウェイクアップされたプロセス $PP-m$ は再度メッセージの受信を試みる。これにより、プロセス $PP-n$ からのメッセージがプロセス $PP-m$ で受信され、異なるプロセッサ相互でプロセス間の通信が実行される。

【0304】

本実施の形態によれば、上記のようにプロセス間通信手段 1000 $P_0 \sim P_n$ によって、セマフォ方式又はメッセージキュー方式を用いた同一プロセッサ内又は異なるプロセッサ間のプロセス通信（同期又はメッセージの送受）を行うこと

により、プロセス切り替えなどのプロセス制御やデータのやり取りを行うことが可能となる。

【0305】

シングルプロセッサ向けOSによるプロセス間通信は、同一プロセッサ内のプロセス間に限られ、異なるプロセッサ間でのプロセス通信は処理の重いネットワーク等を利用して行う必要があったが、ネットワーク通信に比べて処理の高速な制御処理中継手段600とプロセス間通信手段1000P0～Pnによるセマフォ方式又はメッセージキュー方式を用いることで、シングルプロセッサ向けOSを実装するマルチプロセッサシステムにおいても、処理の早いプロセス間通信が実現できる。

【0306】

同一プロセッサ内のプロセス間通信に関しては、シングルプロセッサ向けOSによるプロセスのスリープからウェイクアップに要する処理性能とほぼ同等の性能が得られる。

【0307】

また、マルチプロセッサ向けOSと異なり、プロセッサ毎に異なるシングルプロセッサ向けOSを実装していてもプロセッサ相互でのプロセス間通信が可能となる利点がある。

【0308】

次に、上記プロセス間通信手段1000P0～1000Pnによるセマフォ方式及びメッセージキュー方式を用いたプロセス間通信の具体例について、図27を参照して説明する。

【0309】

図27において、プロセッサP0上でプロセスPP-0が、プロセッサPj上でプロセスPP-jが、プロセッサPk上でプロセスPP-kがそれぞれ実行されている。ここでは、プロセスPP-0がブラウザ、プロセスPP-jがjavaアプレット、プロセスPP-kがMPEG4アプリケーションとして動作し、図28に示すように、プロセスPP-0によるブラウザ1100に埋め込まれたウィンドウ内で、プロセスPP-jによるjavaアプレットのコンテンツ12

00と、プロセスPP-kによるMPEG4映像コンテンツ1300とが同期して表示される。

【0310】

この例では、第1プロセッサ側100のシングルプロセッサ向けOS300P0上で動作するアプリケーションの処理単位のうち、プロセスPP-0をプロセッサP0上で動作させ、アプリケーション内で並列化を行えるプロセスであるプロセスPP-jとプロセスPP-kを第2プロセッサ側200のプロセッサPjとプロセッサPkにおいて並列処理する場合である。

【0311】

図27において、javaアプレット（プロセスPP-j）とMPEG4アプリケーション（プロセスPP-k）は起動されると、共にブラウザ（プロセスPP-0）からのjavaアプリデータ及びMPEG4映像データの受信を待つ（ステップS101、S201）。

【0312】

ここで、プロセッサP0のプロセスPP-0からプロセス間通信手段1000P0によるメッセージ送信を利用して、javaアプリデータ及びMPEG4映像データがメッセージとして送信され、javaアプレット（プロセスPP-j）及びMPEG4アプリケーション（プロセスPP-k）で受信される。

【0313】

これにより、javaアプレット（プロセスPP-j）による動作が開始される（ステップS102）、MPEG4アプリケーション（プロセスPP-k）によるデコード処理が開始される（ステップS202）。

【0314】

この例では、javaアプレット（プロセスPP-j）による動作の方がMPEG4アプリケーション（プロセスPP-k）によるデコード処理より早く終了することが予め分かっており、javaアプレット（プロセスPP-j）は動作終了後に、MPEG4アプリケーション（プロセスPP-k）の処理完了を待つように、セマフォのカウンタの初期値を「0」に設定し、早く終了したjavaアプレット（プロセスPP-j）がセマフォdownできないようになっている

。また、MPEG4アプリケーション（プロセスPP-k）は処理終了後にセマフォupするように定義されている。

【0315】

javaアプレット（プロセスPP-j）は、セマフォdownしようとするが（ステップS103）、カウンタの初期値が「0」であるためにdownできずにスリープされ、セマフォ待ちの状態となる。

【0316】

MPEG4アプリケーション（プロセスPP-k）が処理が終了すると（ステップS203）、セマフォupを要求する（ステップS204）。これにより、セマフォ待ちのjavaアプレット（プロセスPP-j）がウェイクアップされ、セマフォdownが可能となる。

【0317】

以上の動作を、フレーム毎に繰り返すことにより、プロセスPP-0によるブラウザ1100に埋め込まれたウィンドウ内で、プロセスPP-jによるjavaアプレットのコンテンツ1200と、プロセスPP-kによるMPEG4映像コンテンツ1300とが同期して表示される。

【0318】

次に、本発明の第4の実施の形態による並列処理システムについて、図29以下を参照して説明する。図29は、この第4の実施の形態による並列処理システムの構成を示すブロック図であり、図13と共通の構成要素については同じ符号を用いている。

【0319】

図29に示すように、第4の実施の形態による並列処理システムにおいては、システムバス91で接続される複数のプロセッサ（CPU）P1～Pn（nは2以上の整数）からなるマルチプロセッサを構成しており、プロセッサP1～Pnの各プロセッサ上で動作するシングルプロセッサ向けOS300P1～300Pnを実装している。

【0320】

すなわち、マルチプロセッサを、論理的に第1プロセッサ側と第2プロセッサ

側の二組に分ける構成をとっていない点が、第1の実施の形態から第3の実施の形態と相違する。

【0321】

この第4の実施の形態では、上記第3の実施の形態と同様に、プロセッサP1～Pnの並列処理を行うための並列処理通信手段400P1～400Pnと制御代理手段500P1～500Pnに加えて、プロセッサP1～Pn上で実行される各プロセス間での通信を実現するためのプロセス間通信手段1000P1～1000Pnを備えている。

【0322】

一方、第1の実施の形態のようなOSサービス手段50P1～50Pnや、第1の実施の形態～第3の実施の形態のような並列処理通信手段400P1～400Pnを備えることなく、各プロセッサP1～Pn上のプロセス制御が可能である。また、各プロセッサP1～Pn上のシングルプロセッサ向けOS300P1～300Pnについては、同じOSである必要はなく互いに異なってもよい。

【0323】

すなわち、本実施の形態でも、これまでマルチプロセッサ向けOSに実装されていたプロセス間通信の機能を、マルチプロセッサ上でシングルプロセッサ向けOS及びアプリケーションを動作させるシングルプロセッサ向けOSによる並列処理システム上で実現するものであり、ユーザレベルによるプロセス間通信機能の実装を可能にしている。

【0324】

この各プロセッサP1～プロセッサPn内における各プロセスの実行については、他のプロセッサとの排他制御を必要とすることなく行われる。

【0325】

個々のプロセッサP1～プロセッサPn内におけるプロセス間通信及びプロセッサ相互におけるプロセス間通信については、第3の実施の形態で説明したように、プロセス間通信手段1000P1～1000Pnによるセマフォ方式及びメッセージキュー方式を用いて行われ、プロセス相互の同期処理及びデータの受け渡しが実行される。

【0326】

なお、上記各実施の形態の並列処理システムは、並列処理手段、OSサービス手段、制御処理中継手段、代理手段、プロセス間通信手段の各機能を有する並列処理プログラムによって実現することができる。この並列処理プログラムは、磁気ディスク、半導体メモリその他の記録媒体に格納され、その記録媒体からコンピュータ処理装置にロードされ、コンピュータ処理装置の動作を制御することにより、上述した各機能を実現する。

【0327】

以上好ましい実施の形態及び実施例をあげて本発明を説明したが、本発明は必ずしも上記実施の形態及び実施例に限定されるものではなく、その技術的思想の範囲内において様々に変形して実施することができる。

【0328】**【発明の効果】**

以上説明したように本発明によれば、マルチプロセッサ上でシングルプロセッサ向けOSと既存のアプリケーションをそれらに改造を加えることなく動作させ、アプリケーションに対してマルチプロセッサによる並列処理を実現できる。

【0329】

また、本発明によれば、並列処理手段、OSサービス手段、制御処理中継手段及び代理手段を、モジュールとして実装することにより、シングルプロセッサ向けOS上で実行されるアプリケーションに改造を加えることなく、マルチプロセッサのシステム構成上で、シングルプロセッサ向けOS及び既存のアプリケーションをオーバーヘッドなく動作させることができると共に、マルチプロセッサによる並列処理の恩恵を受けることが可能となる。

【0330】

さらに、第2プロセッサ側に個別にファイルアクセスのための処理手段を設けることなく、第2プロセッサ側におけるタスクはシングルプロセッサ向けOSのサービスをそのまま利用してファイルアクセスが可能となる。また、これにより、ファイルアクセスのための排他処理については、第1プロセッサ側で完結するため、マルチプロセッサ向けOS上でアプリケーションを動作させる場合のよう

な排他処理によるオーバヘッドが発生することなく並列処理を実現できる。

【0331】

また、第1プロセッサ側のプロセッサ及び第2プロセッサ側の各プロセッサに、仮想記憶機構を備えるシングルプロセッサ向けOSを搭載することにより、各プロセス間のメモリ保護を行いながら上述した並列処理を実現することができる。

【0332】

また、第1プロセッサ側の処理単位（タスク／プロセス）あるいは第2プロセッサ側の処理単位的一方ないし双方が、並列処理手段に対して処理を要求する際、並列処理手段もしくは第2プロセッサ側のシングルプロセッサ向けOSが、要求元処理単位を待機状態とすることにより、同期的な処理が実現できる。

【0333】

さらに、第1プロセッサ側の処理単位（タスク／プロセス）あるいは第2プロセッサ側の処理単位的一方ないし双方が、並列処理手段に対して処理を要求する際、並列処理手段もしくは第2プロセッサ側のシングルプロセッサ向けOSが、要求元処理単位の実行を継続させることにより、非同期的な処理を実現することができる。

【0334】

さらに、ユーザレベルでプロセス間の通信機能を実装することにより、マルチプロセッサ向けOSを必要とすることなく、シングルプロセッサ向けOS上でも、並列プロセッサでのプロセス間の通信が可能となる。

【0335】

また、マルチプロセッサ向けOSと異なり、プロセッサ毎に異なるシングルプロセッサ向けOSを実装していてもプロセッサ相互でのプロセス間通信が可能となる利点が得られる。

【図面の簡単な説明】

【図1】 本発明の第1の実施の形態による並列処理システムの構成を示すブロック図である。

【図2】 第1の実施の形態による並列処理システムの並列処理手段による並列処理起動の動作を説明するための図である。

【図 3】 並列処理システムの並列処理手段による並列処理起動における要求を行うためのメッセージの構成を示す図である。

【図 4】 第 1 の実施の形態による並列処理システムにおける第 2 プロセッサ側のタスクからのファイルアクセス要求に対する OS サービス手段の動作を説明する図である。

【図 5】 第 1 プロセッサ側の逐次タスクからのファイルアクセス要求に対する動作を説明する図である。

【図 6】 第 1 の実施の形態による並列処理システムにおける制御処理中継手段の内部構成を示すブロック図である。

【図 7】 第 1 の実施の形態による並列処理システムにおける制御処理中継手段の処理動作を説明する図である。

【図 8】 第 1 の実施の形態による並列処理システムにおける代理手段によるシグナル通知処理動作を説明する図である。

【図 9】 第 1 の実施の形態による並列処理システムにおける第 2 プロセッサ側におけるタスク相互の連携動作を実現するための構成を示すブロック図である。

【図 10】 第 1 の実施の形態による並列処理システムにおける第 2 プロセッサ側におけるタスク相互の連携動作を説明する図である。

【図 11】 並列処理システムにおける適用例としてのマルチプロセッサの携帯端末においてアプリケーションを動作させる場合の例を説明する図である。

【図 12】 本発明の第 2 の実施の形態による並列処理システムの構成を示すブロック図である。

【図 13】 第 2 の実施の形態による並列処理システムにおけるプロセスの同期的な並列処理起動の動作を説明するための図である。

【図 14】 第 2 の実施の形態による並列処理システムにおけるプロセスの非同期的な並列処理起動の動作を説明するための図である。

【図 15】 第 2 の実施の形態による並列処理システムにおける制御処理中継手段の内部構成を示すブロック図である。

【図 16】 第 2 の実施の形態による並列処理システムにおける制御処理中

継手段の処理動作を説明する図である。

【図 17】 本発明の第 3 の実施の形態による並列処理システムの構成を示すブロック図である。

【図 18】 第 3 の実施の形態による並列処理システムにおいて、プロセッサ内のプロセスがプロセス間通信手段によるセマフォ方式を用いてセマフォ up 又は down を行う場合の動作を説明するための図である。

【図 19】 第 3 の実施の形態による並列処理システムにおいて、プロセッサ内のプロセスがプロセス間通信手段によるセマフォ方式を用いてセマフォ down を行う場合の動作を説明するための図である。

【図 20】 第 3 の実施の形態による並列処理システムにおいて、同一プロセッサ内のプロセス間でプロセス間通信手段によるセマフォ方式を用いてセマフォ up を行う場合の動作を説明するための図である。

【図 21】 第 3 の実施の形態による並列処理システムにおいて、異なるプロセッサ相互のプロセス間でプロセス間通信手段によるセマフォ方式を用いてセマフォ up を行う場合の動作を説明するための図である。

【図 22】 第 3 の実施の形態による並列処理システムにおいて、プロセッサ内のプロセスがプロセス間通信手段によるメッセージキュー方式を用いてメッセージの受信を行う場合の動作を説明するための図である。

【図 23】 第 3 の実施の形態による並列処理システムにおいて、プロセッサ内のプロセスがプロセス間通信手段によるメッセージキュー方式を用いてメッセージの受信を行う場合の動作を説明するための図である。

【図 24】 第 3 の実施の形態による並列処理システムにおいて、プロセッサ内のプロセスがプロセス間通信手段によるメッセージキュー方式を用いてメッセージの送信を行う場合の動作を説明するための図である。

【図 25】 第 3 の実施の形態による並列処理システムにおいて、同一プロセッサ内のプロセス間でプロセス間通信手段によるメッセージキュー方式を用いてメッセージの送受信を行う場合の動作を説明するための図である。

【図 26】 第 3 の実施の形態による並列処理システムにおいて、異なるプロセッサ相互のプロセス間でプロセス間通信手段によるメッセージキュー方式を

用いてメッセージの送受信を行う場合の動作を説明するための図である。

【図 27】 第3の実施の形態による並列処理システムにおいて、セマフォ方式及びメッセージキュー方式を用いたプロセス間通信の具体例を説明するための図である。

【図 28】 図 27 の具体例におけるブラウザ画面の表示内容を説明する図である。

【図 29】 本発明の第4の実施の形態による並列処理システムの構成を示すブロック図である。

【符号の説明】

- 10、100：第1プロセッサ側
- 20、200：第2プロセッサ側
- 30、300P0、300P1～300Pn：シングルプロセッサ向けOS
- 40P0～40Pn：並列処理手段
- 400P0～400Pn：並列処理通信手段
- 50P0～50Pn：OSサービス手段
- 500P0～500Pn：制御代理手段
- 60、600：制御処理中継手段
- 61P0～61Pn、601P0～601Pn：割込制御装置
- 61a、601a：割込指示部
- 61b、601b：割込状態保持部
- 61c、601c：割込取消部
- 62P0～62Pn、602P0～602Pn：通信領域
- 62a：通信理由保持領域
- 62b：通信データ保持領域
- 62c、602b：排他制御領域
- 602a：通信キュー
- 70：代理手段
- 91：システムバス
- 92：主記憶装置

93: 外部記憶装置

1000P0 ~ 1000Pn: プロセス間通信手段

ST: 逐次タスク

PT: 並列化タスク

PT-1 ~ PT-n: タスク

41P1 ~ 41Pn: 相互排他手段

43 排他制御領域

44: タスク管理内容保持領域

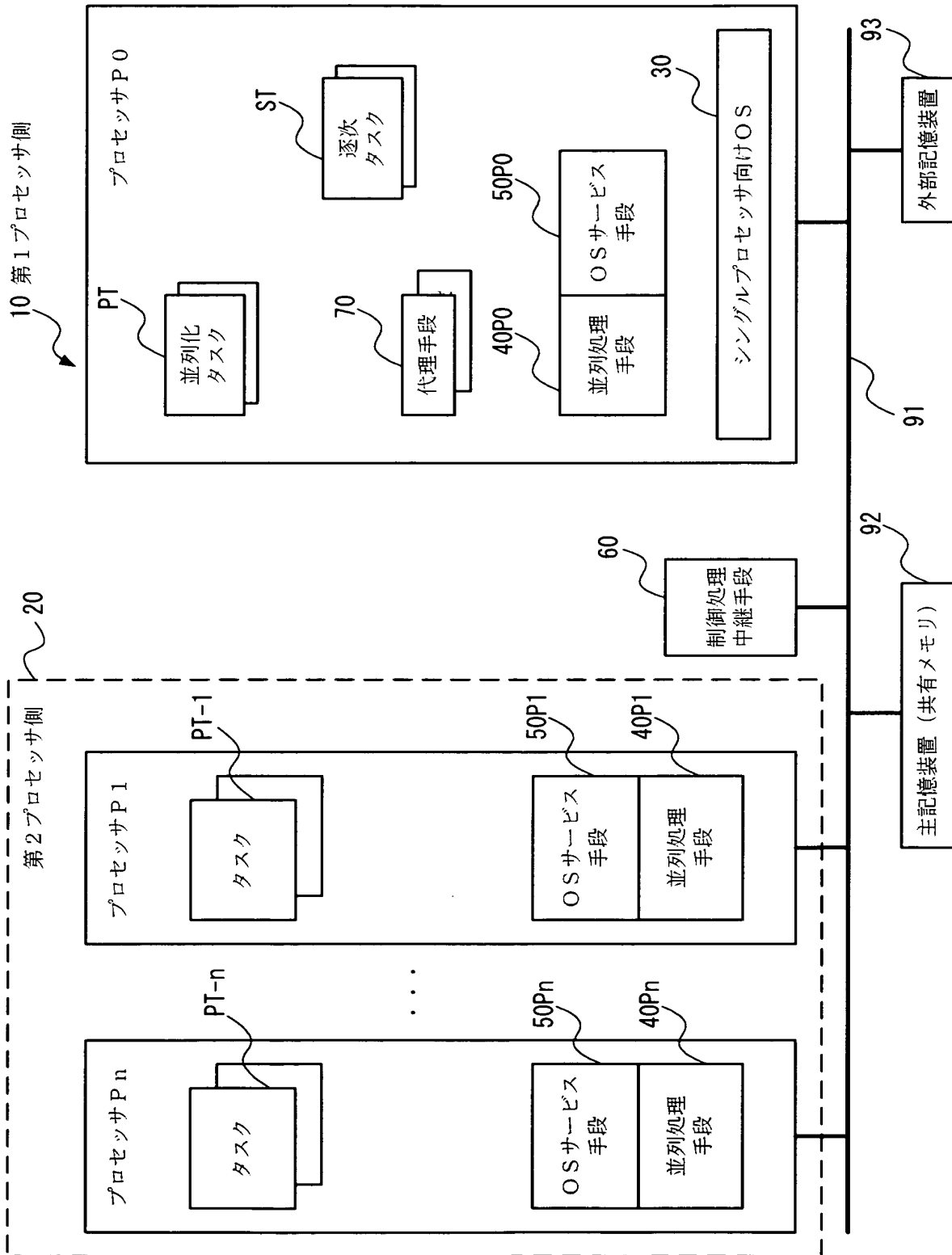
SP: 逐次プロセス

PP: 並列化プロセス

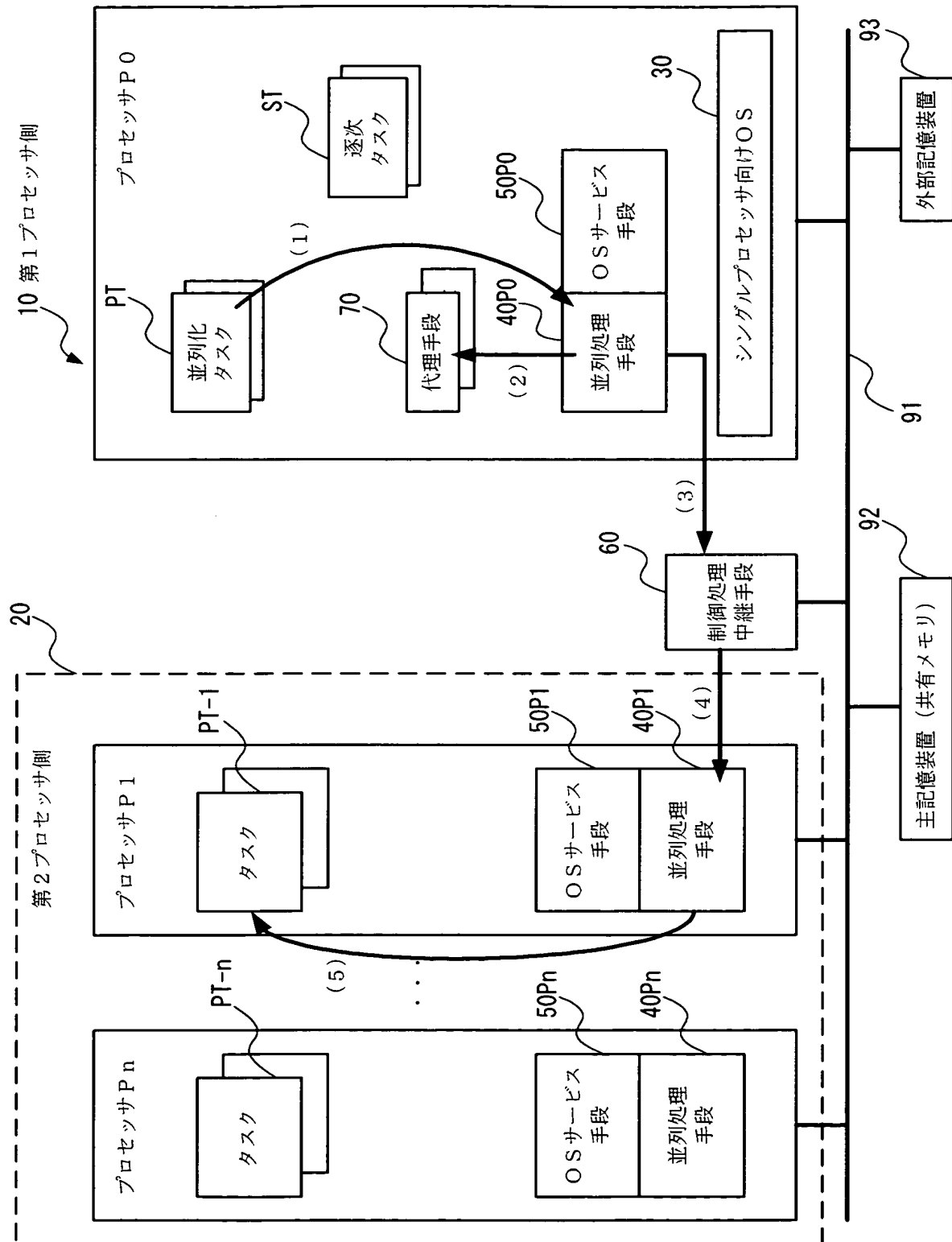
PP-1 ~ PP-n: プロセス

【書類名】 図面

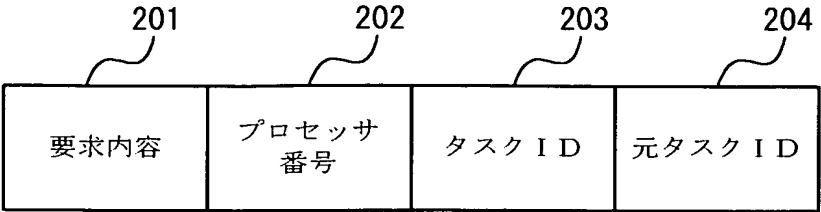
【図 1】



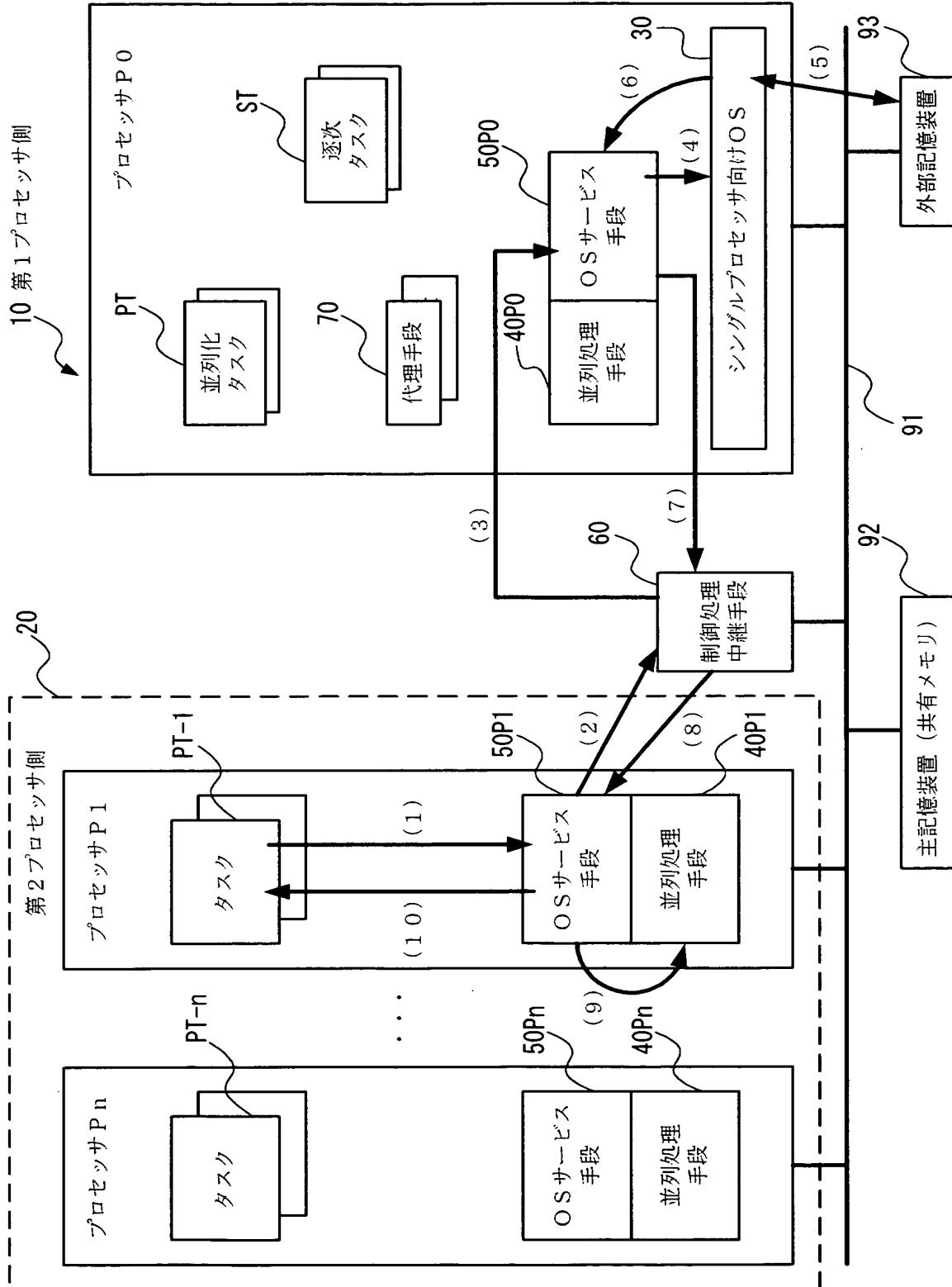
【図 2】



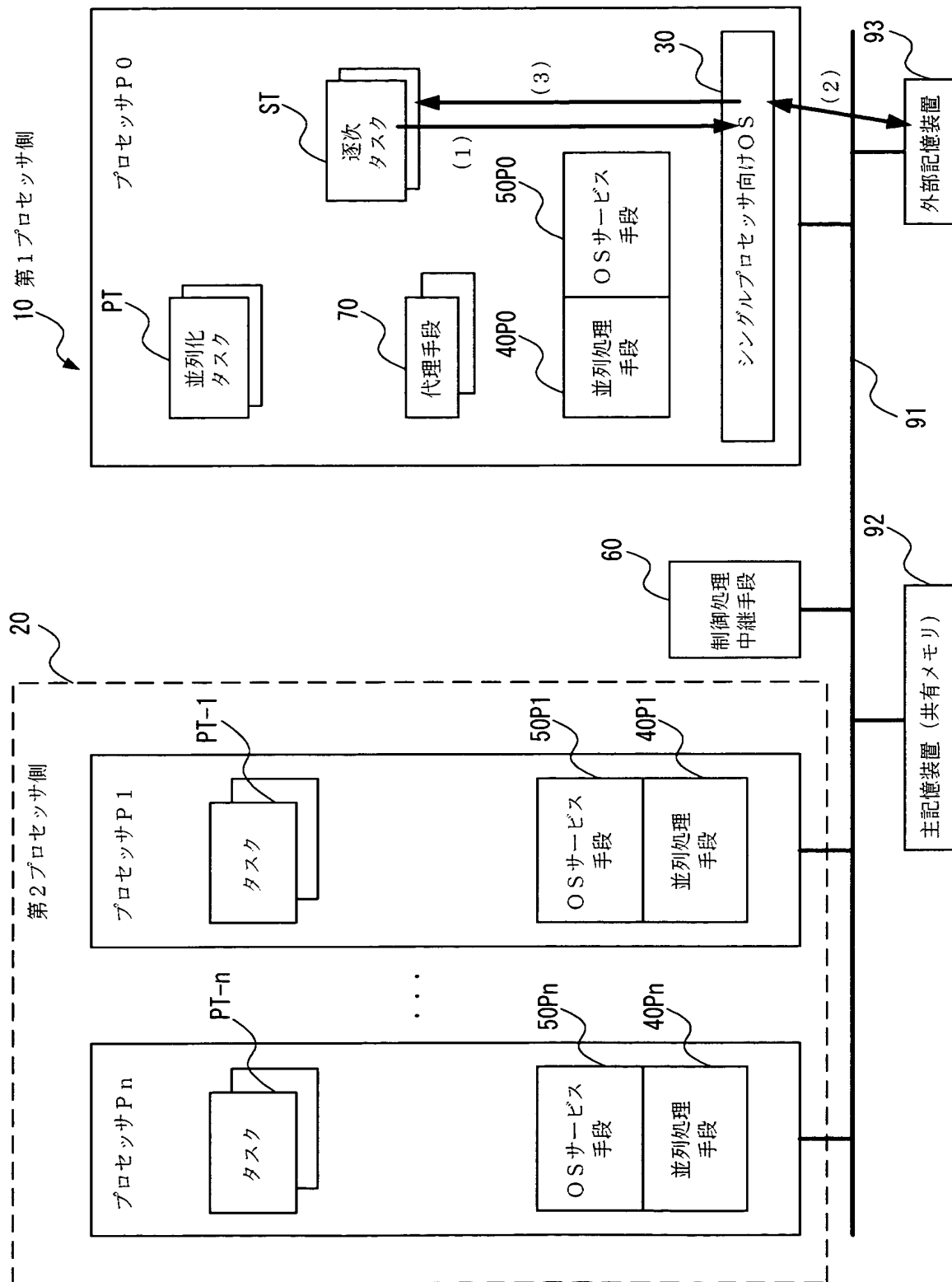
【図 3】



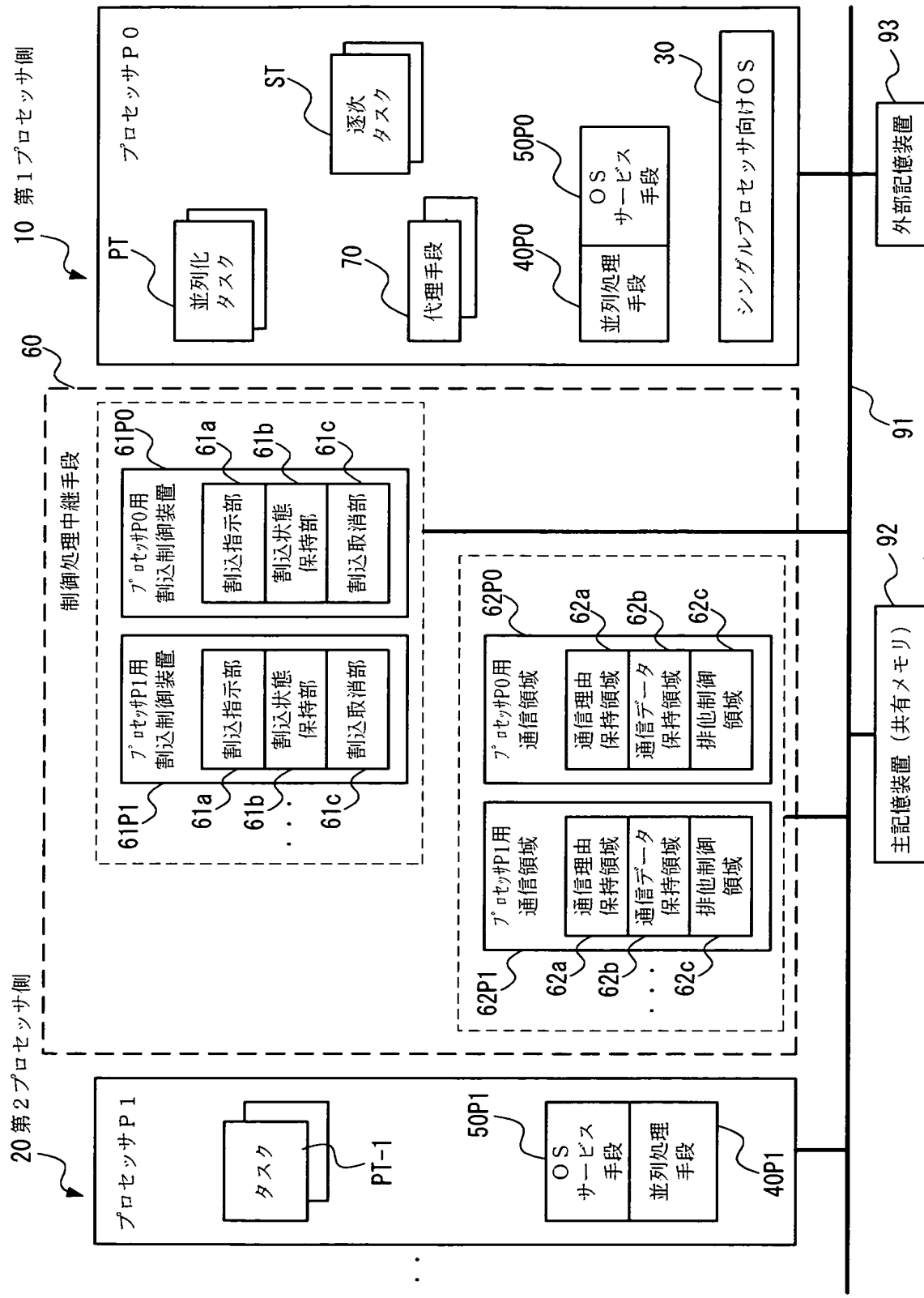
【図 4】



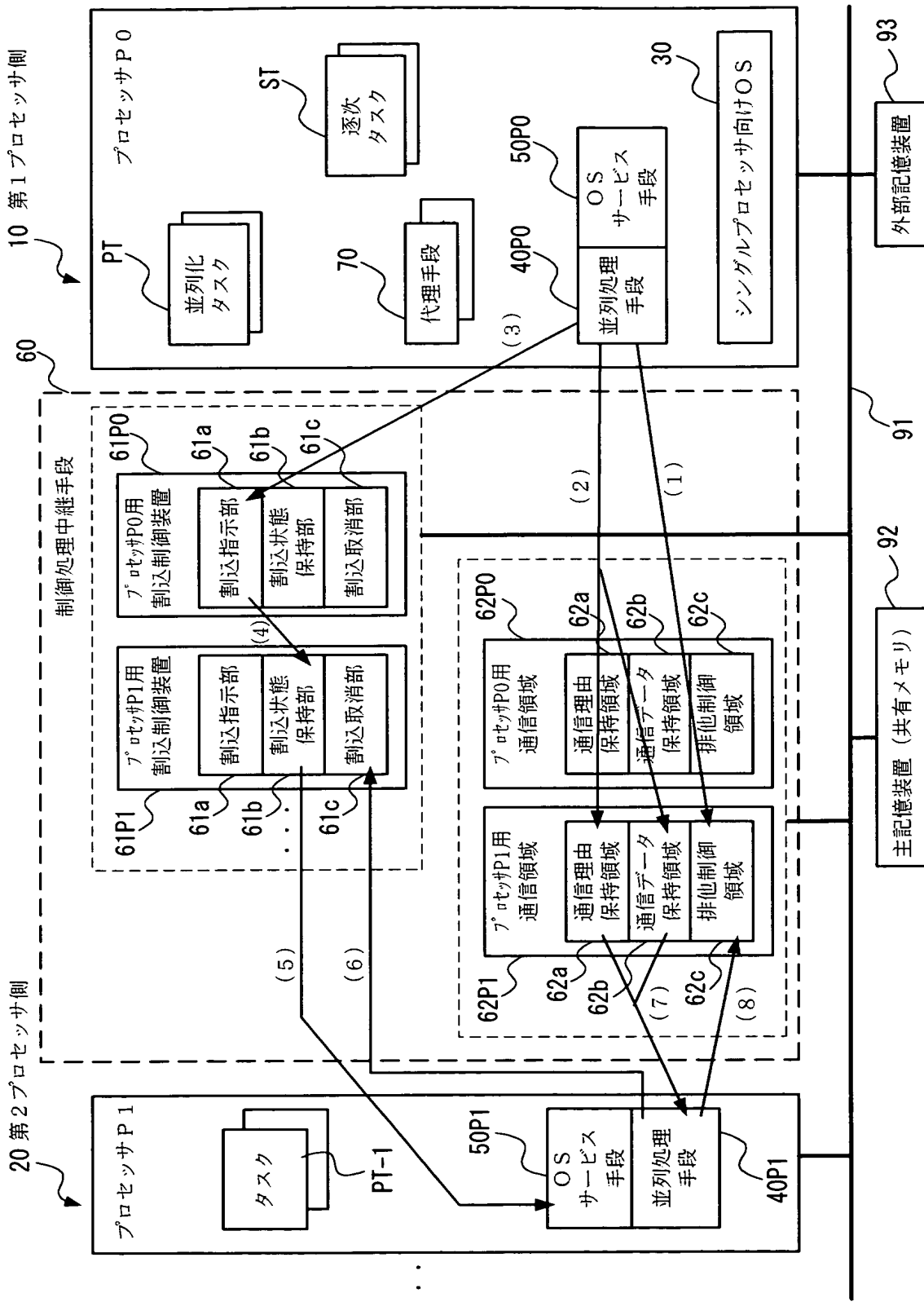
【図 5】



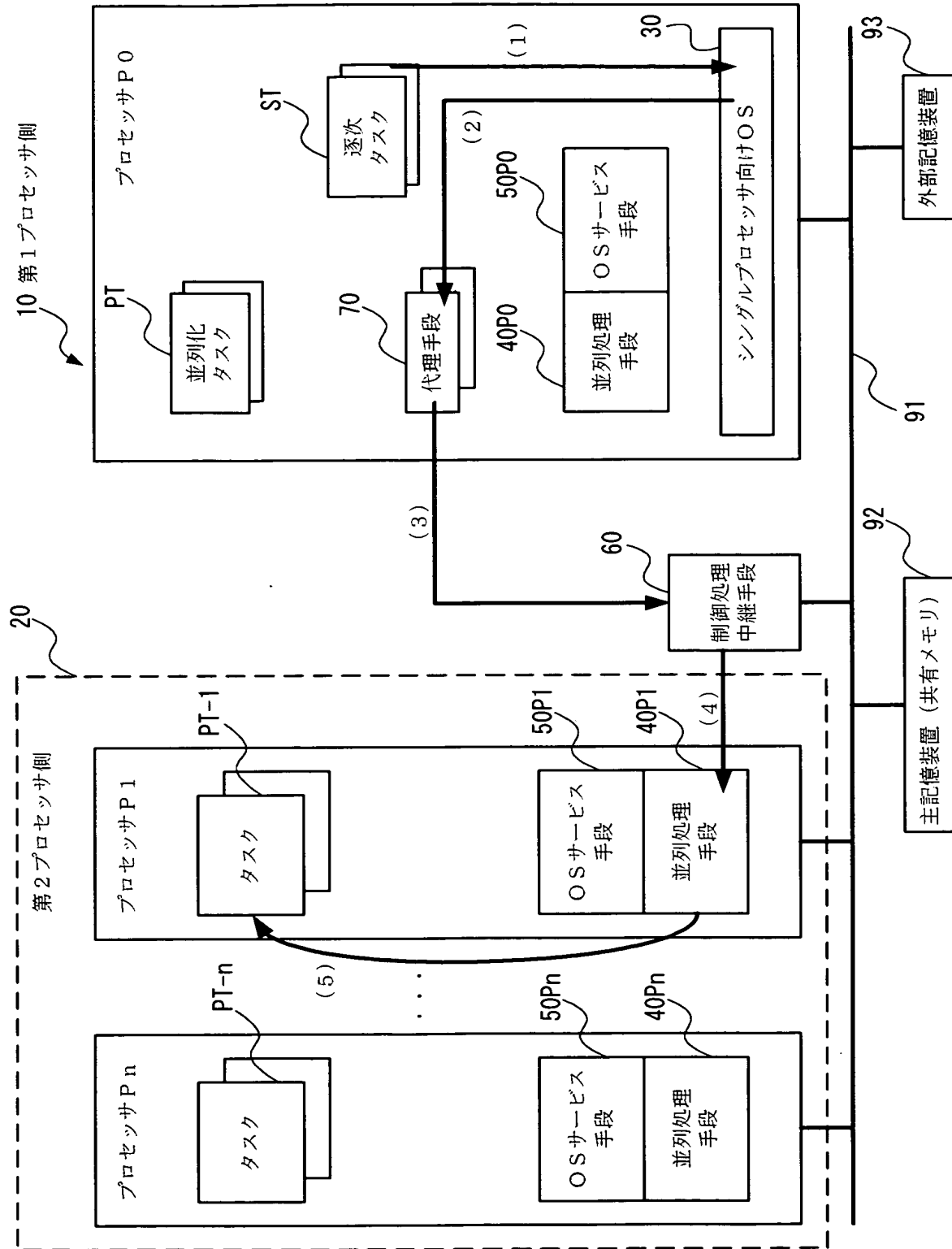
【図6】



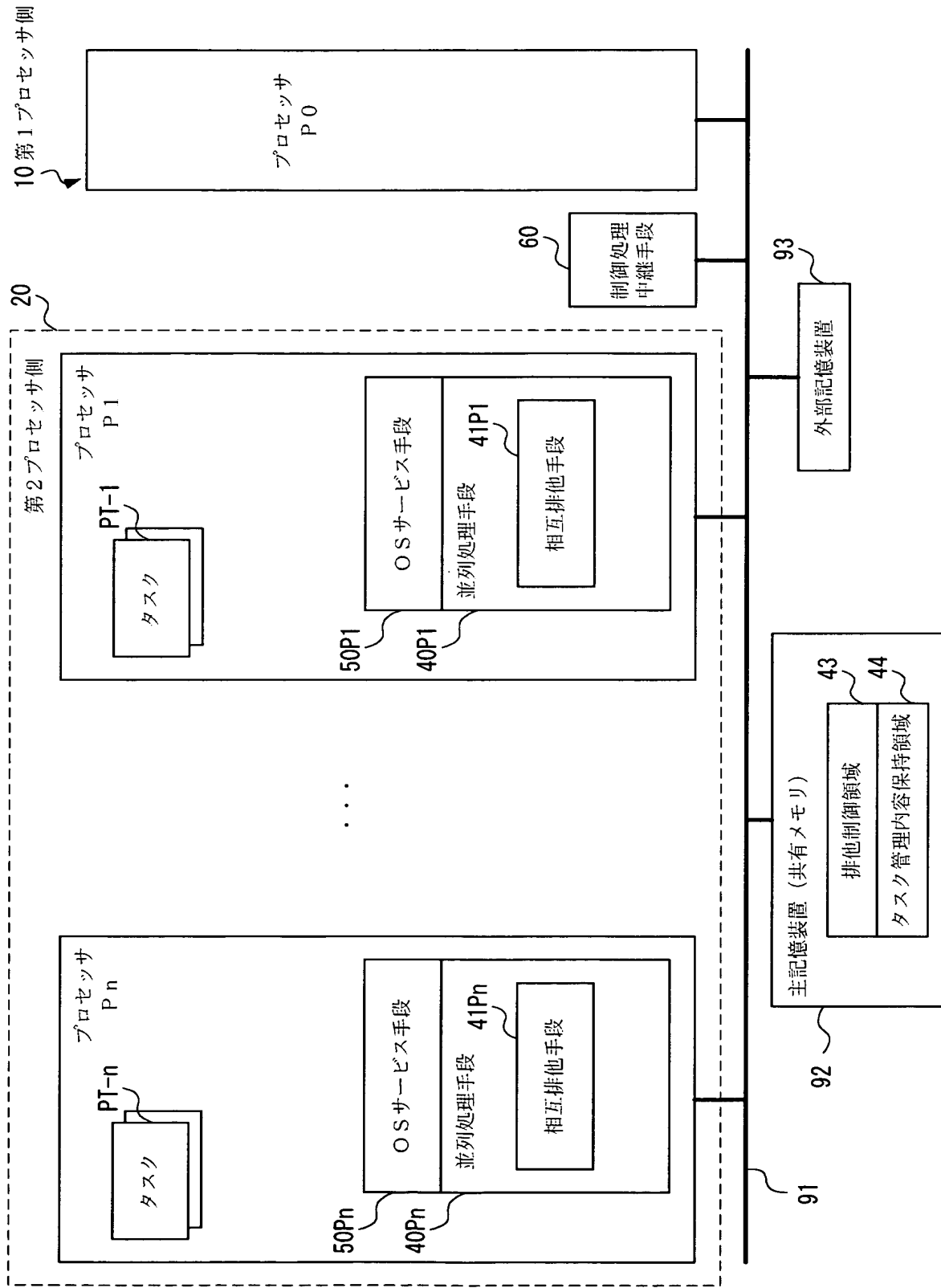
【図7】



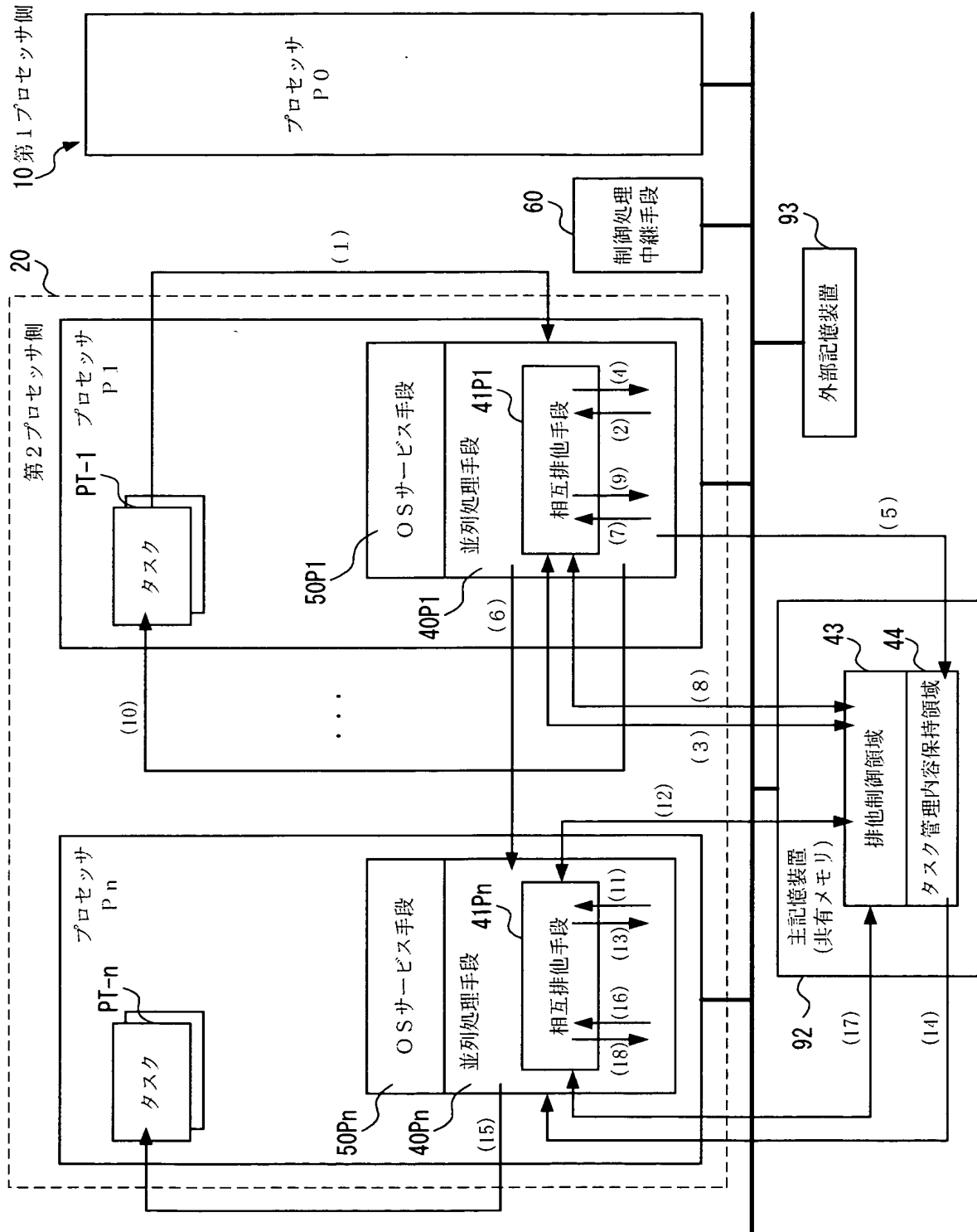
【図 8】



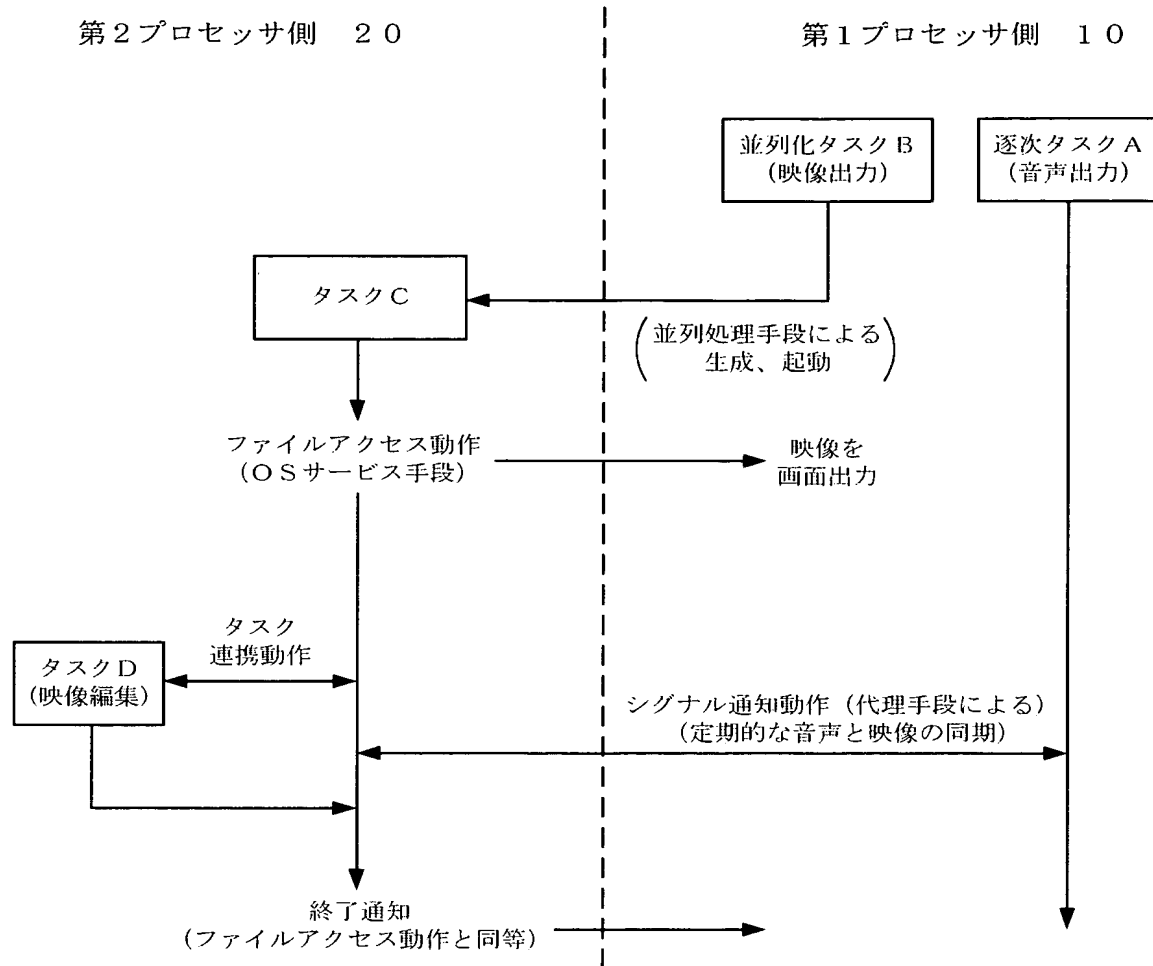
【図 9】



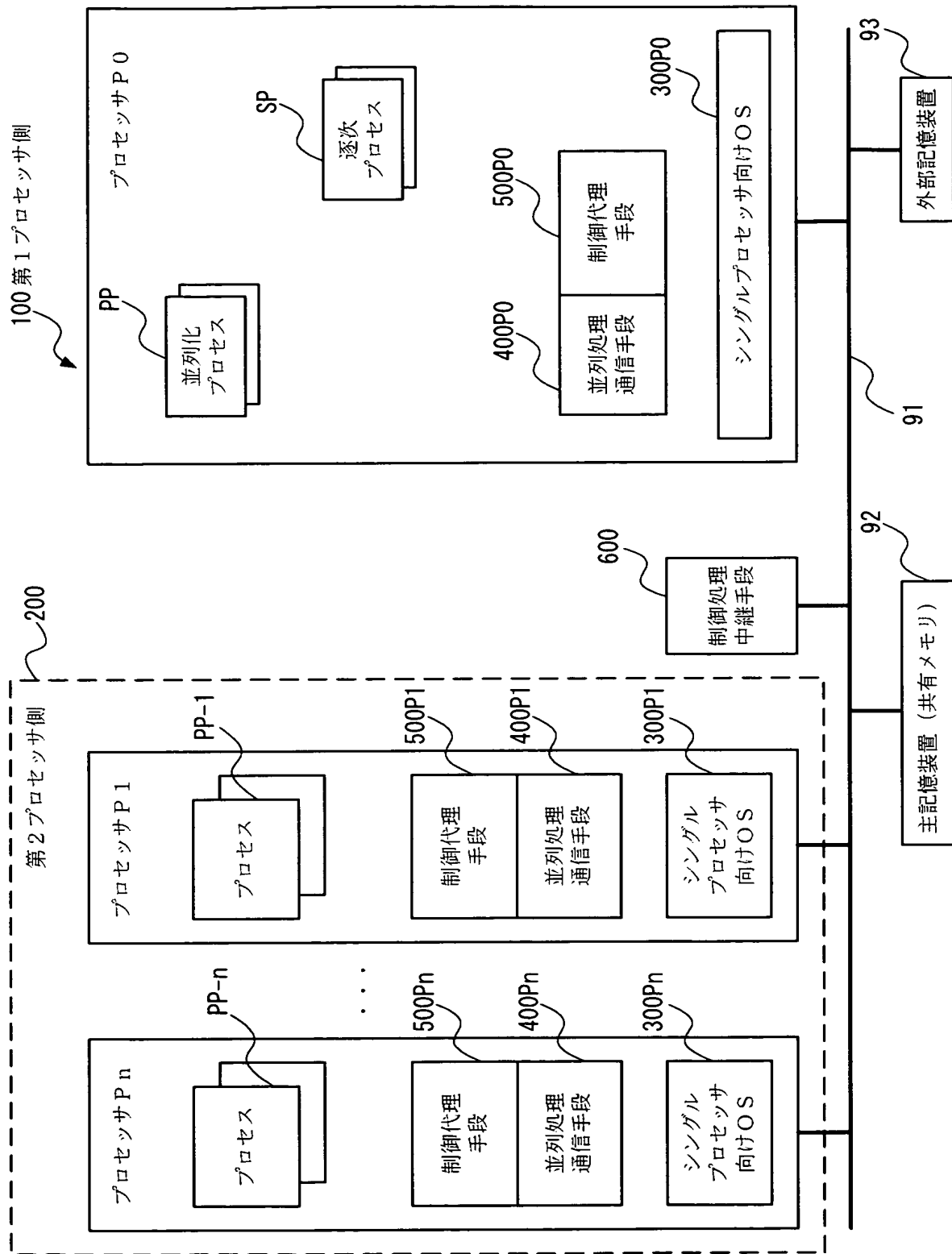
【図 10】



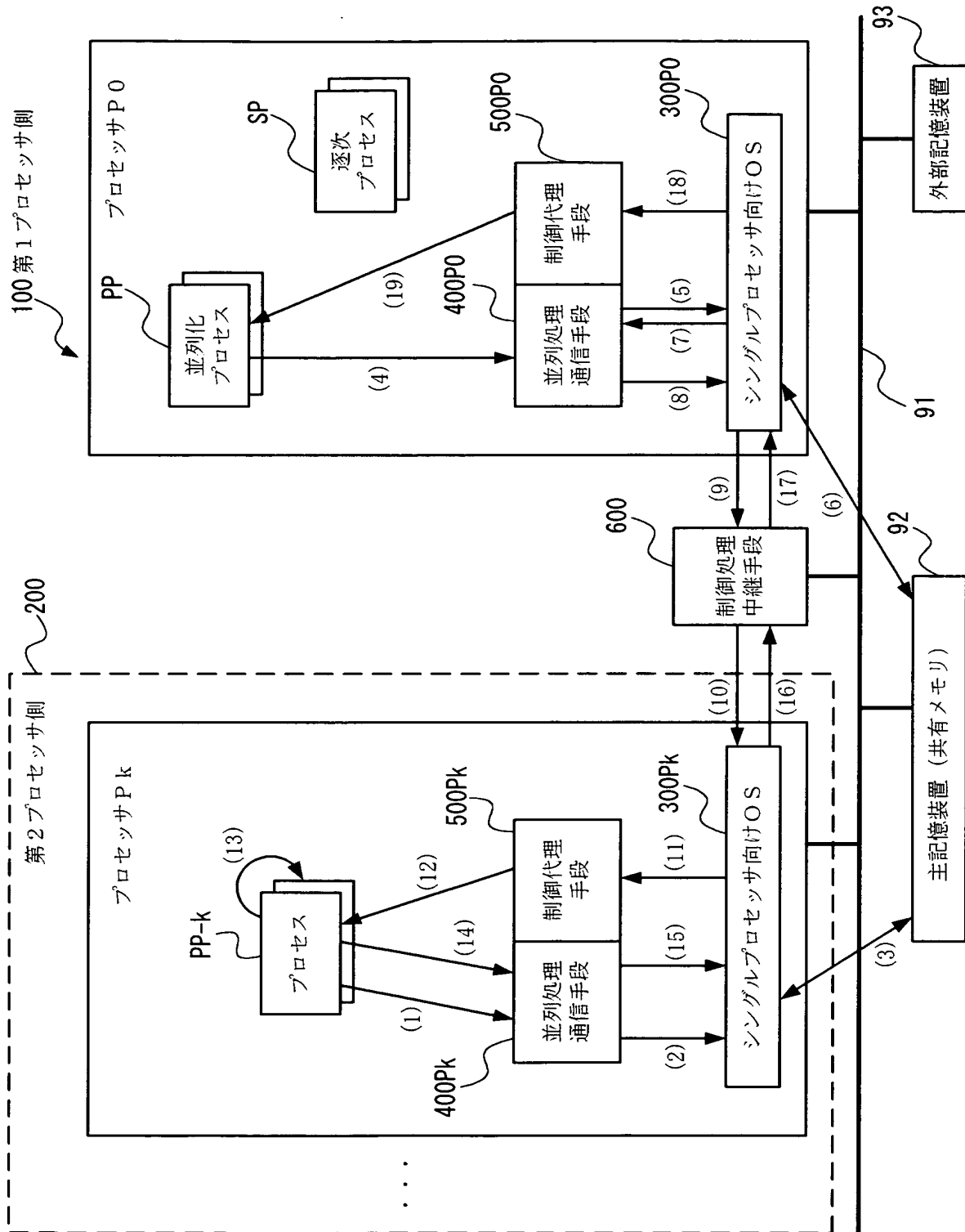
【図 11】



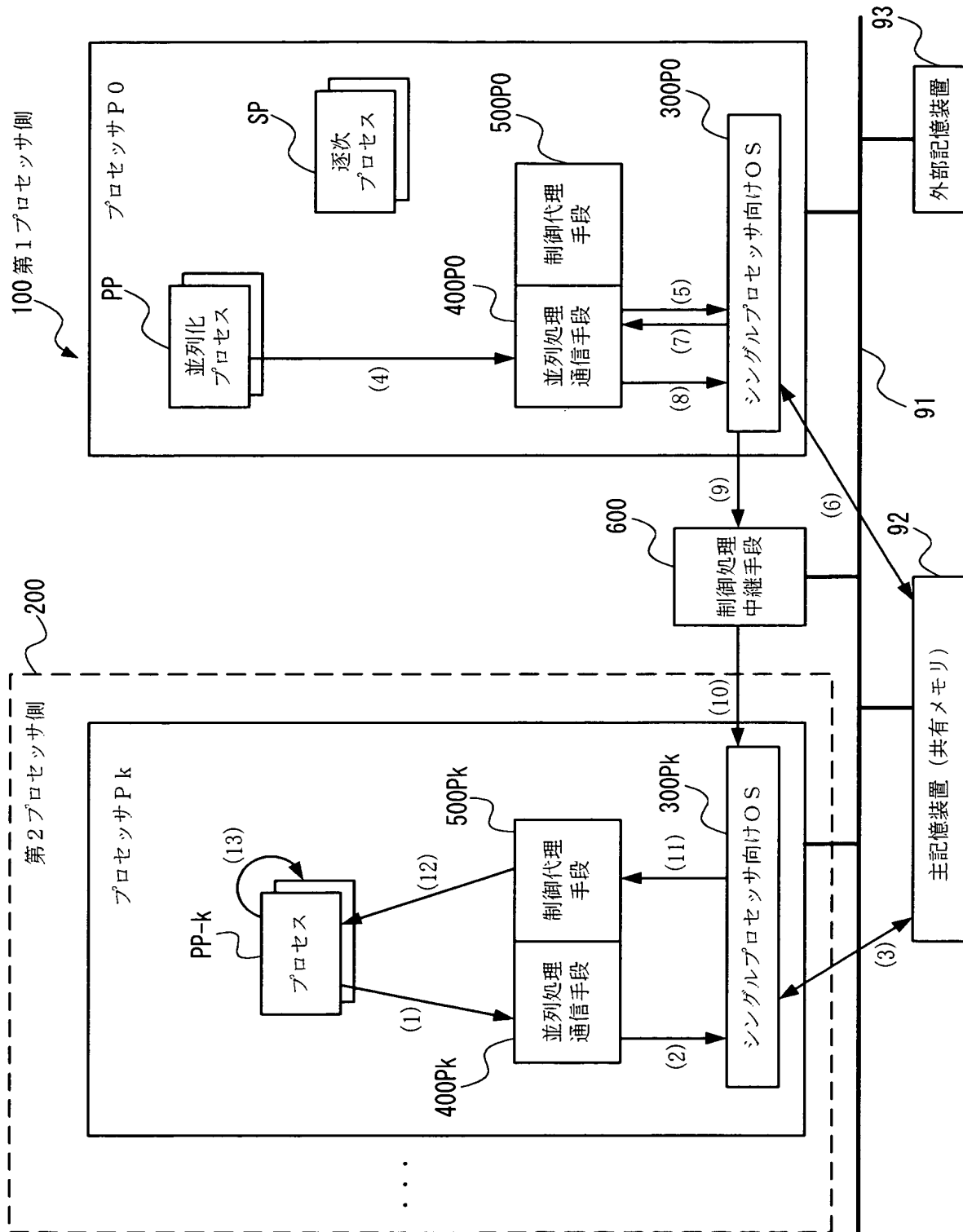
【図 12】



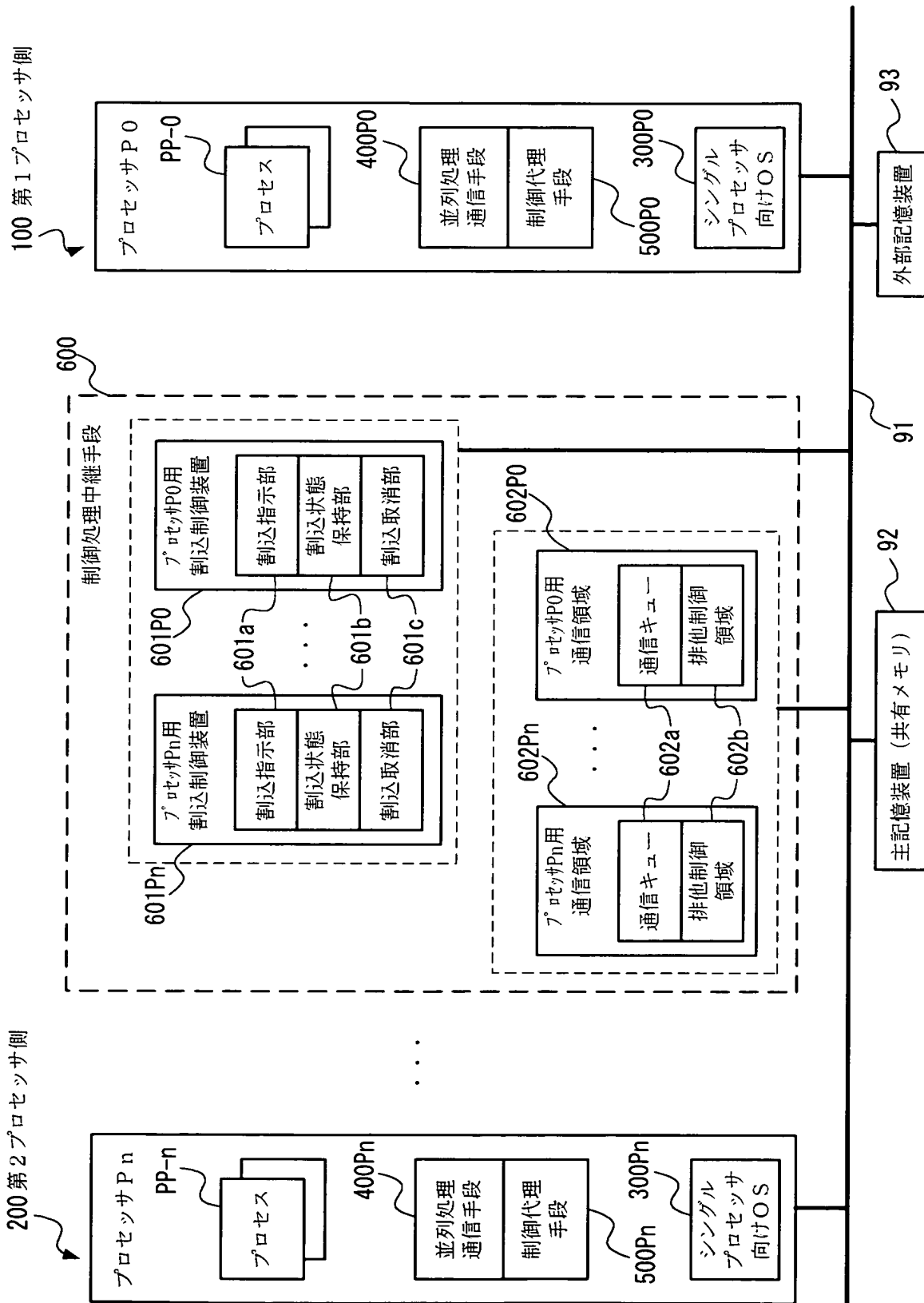
【図 13】



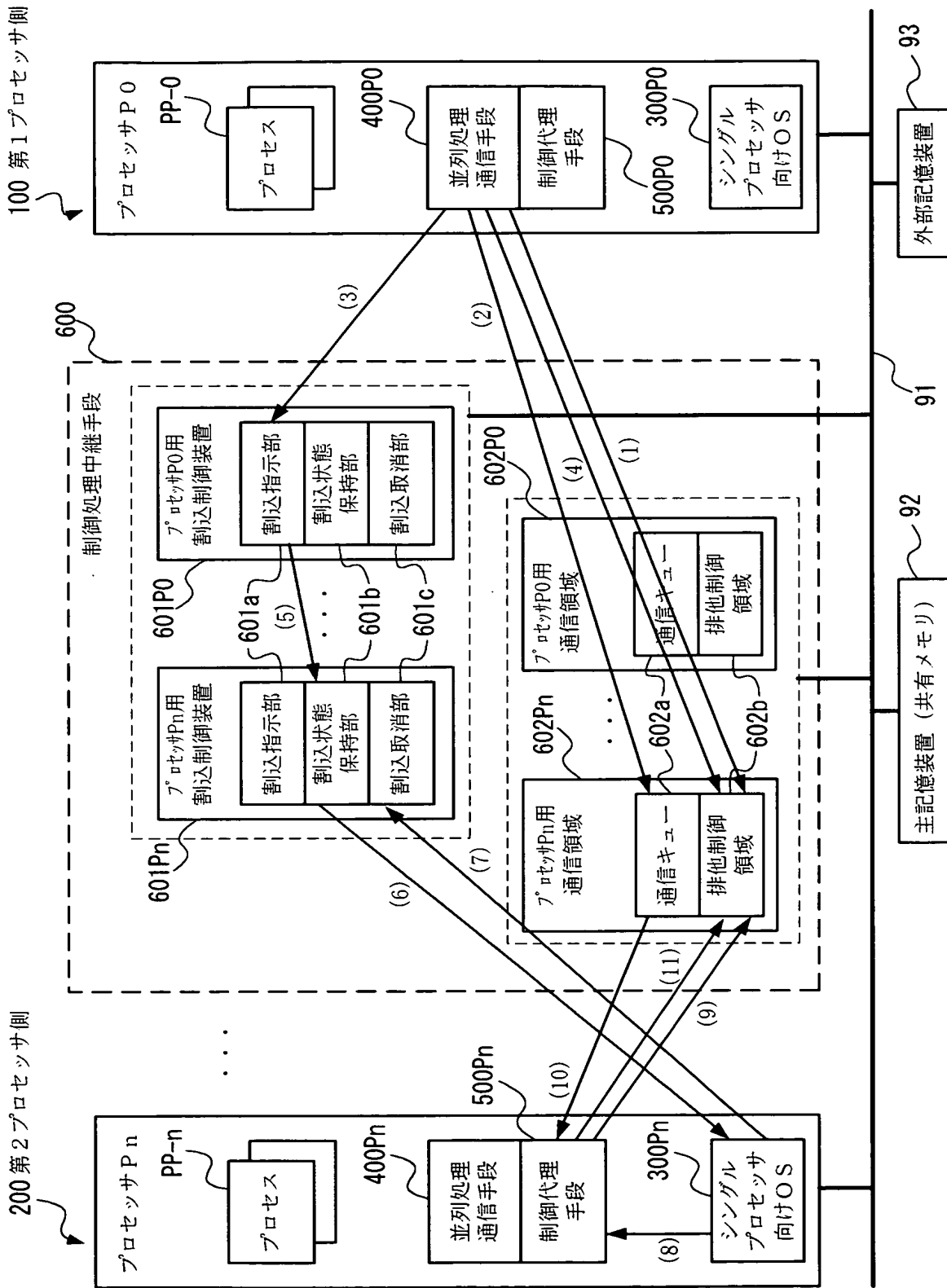
【図 14】



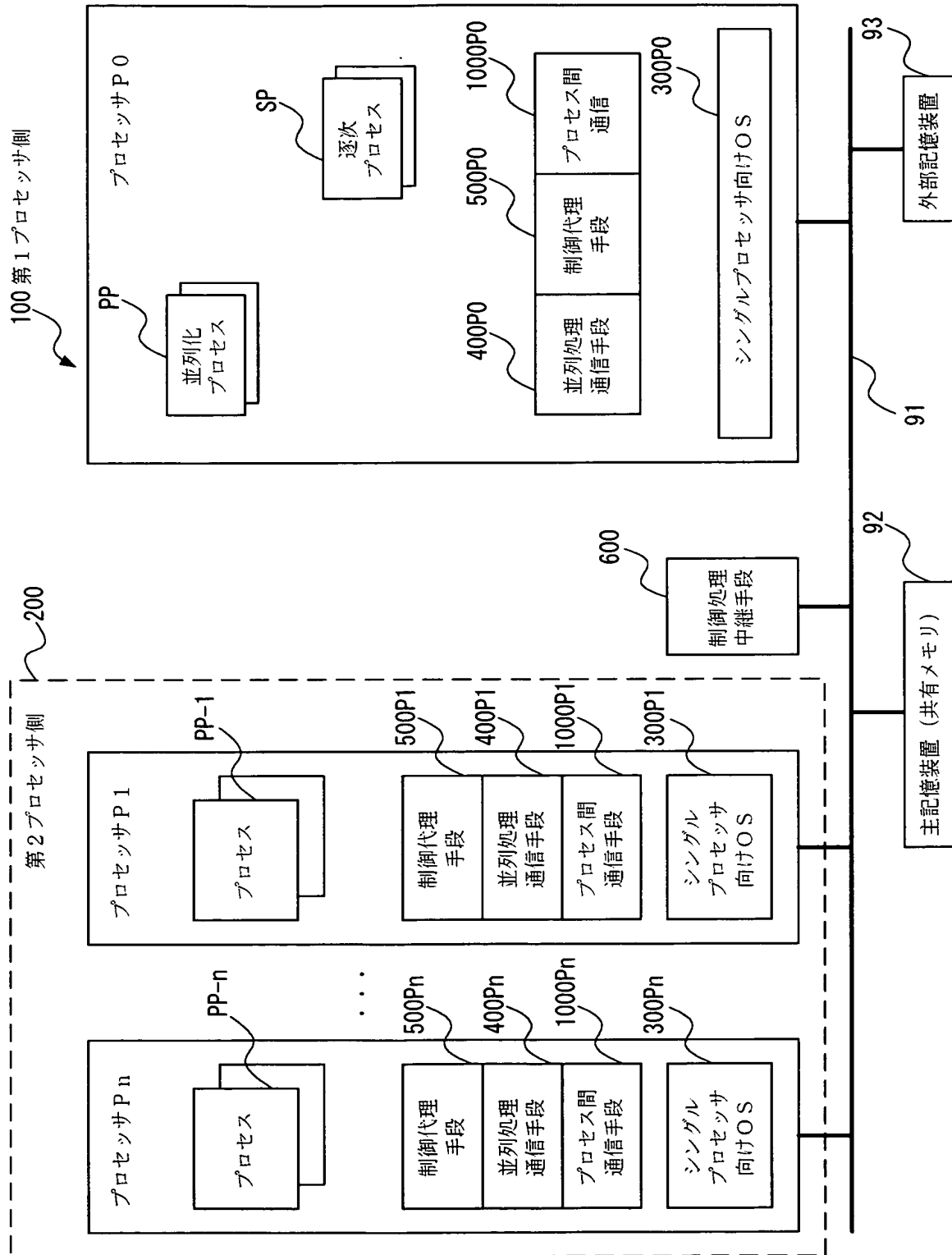
【図 15】



【図 16】

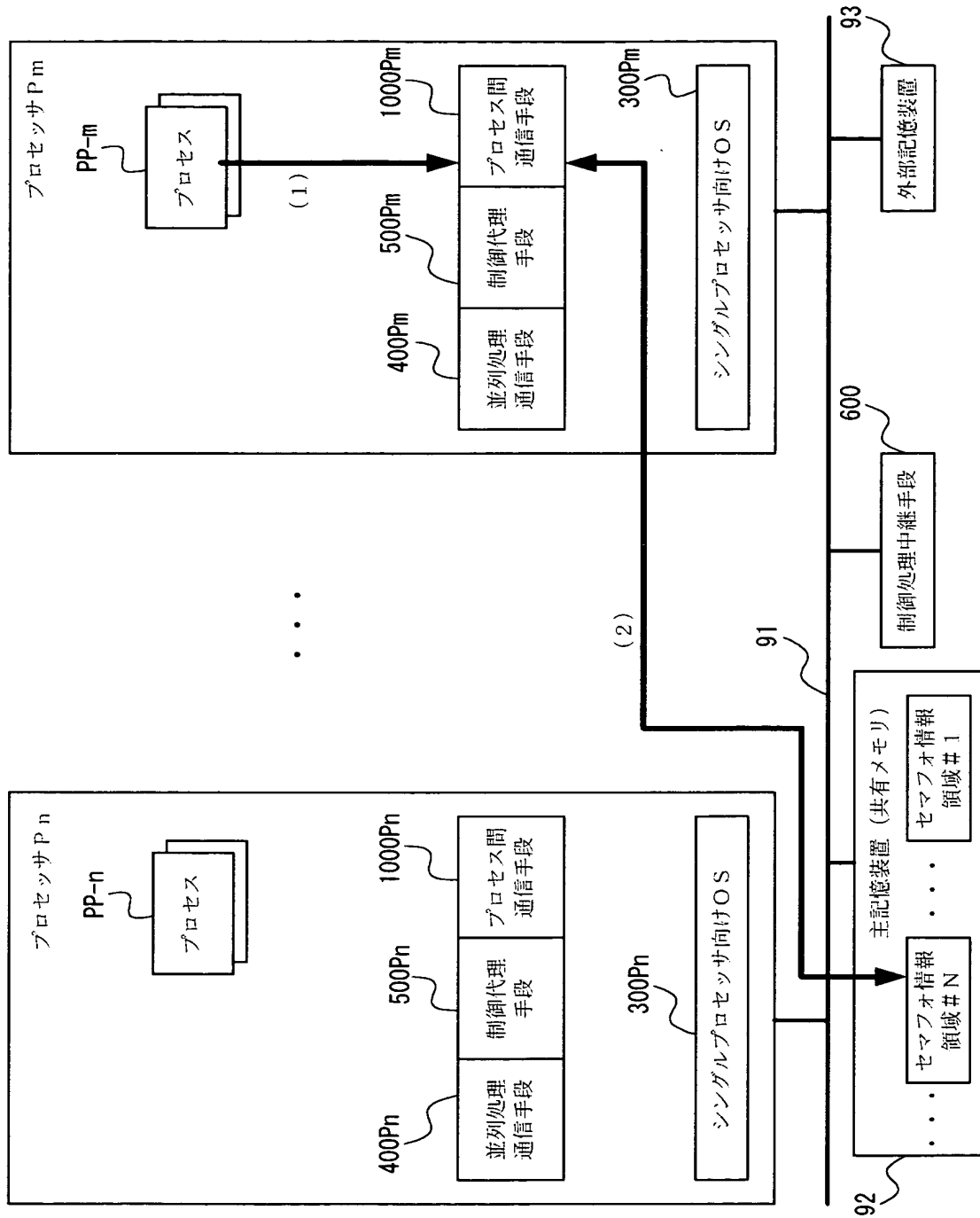


【図 17】



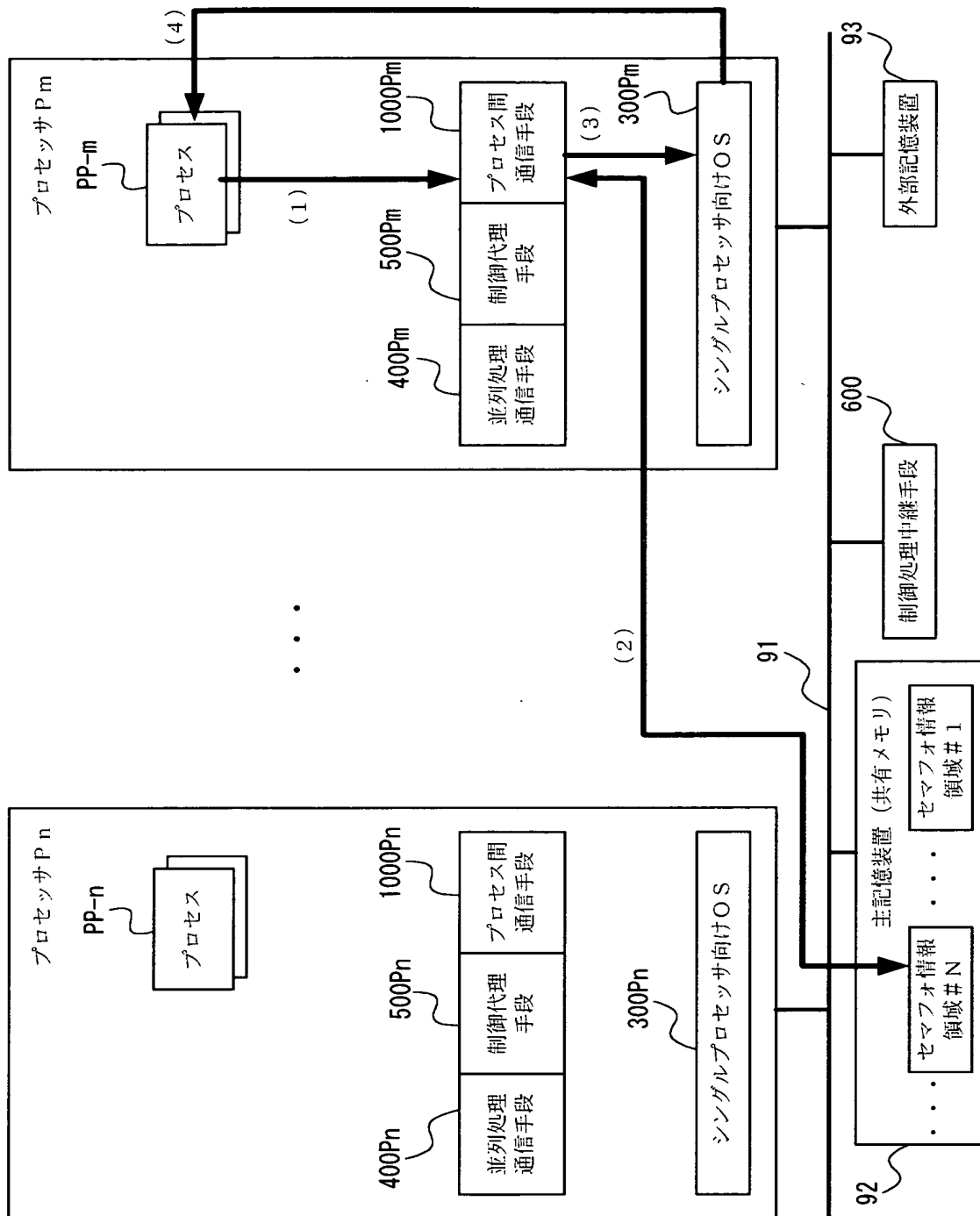
【図 18】

(セマフォ方式による up/down 動作)



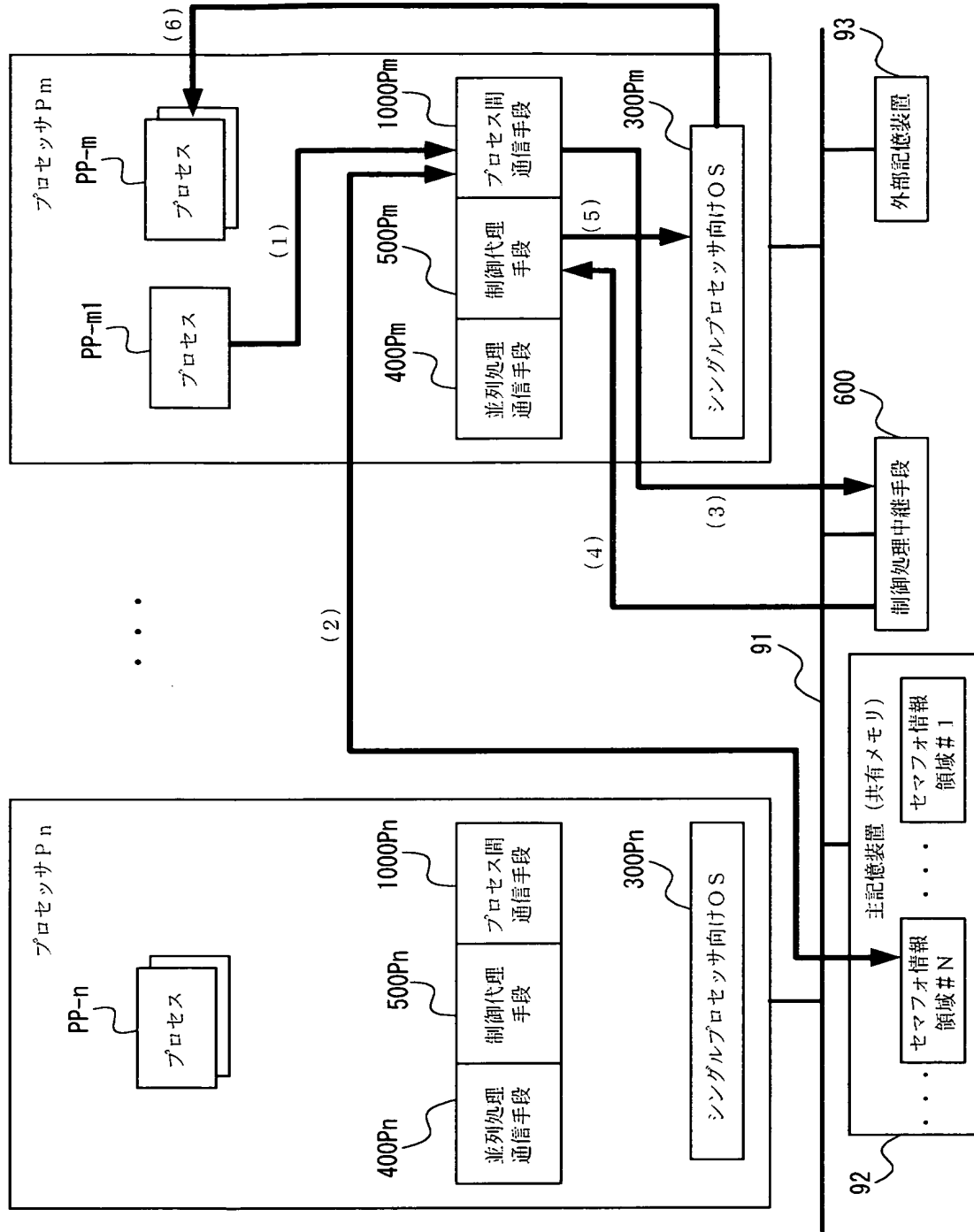
【図 19】

(セマフォによるdown動作)



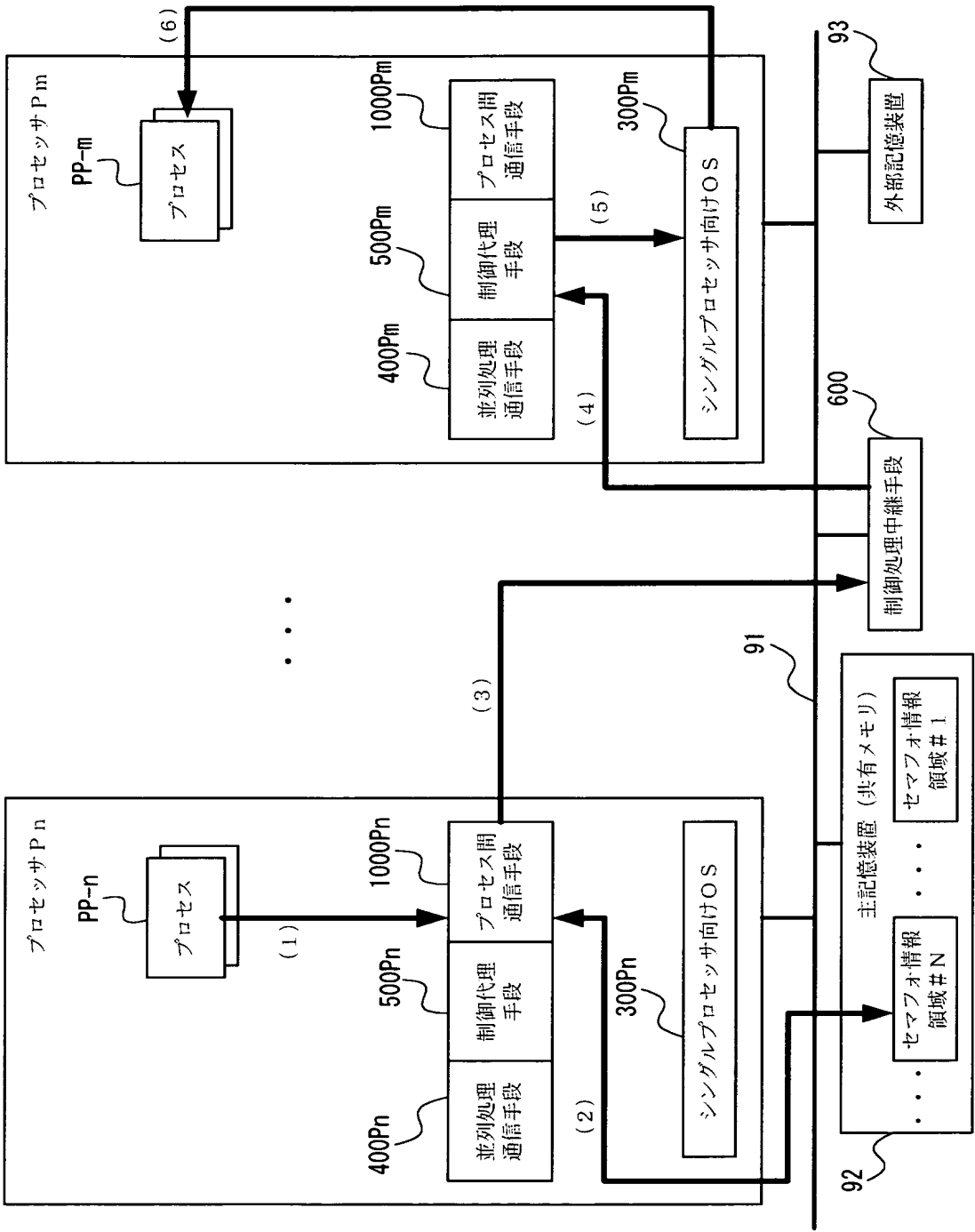
【図 20】

(同一プロセッサにおけるセマフォ方式によるup動作)



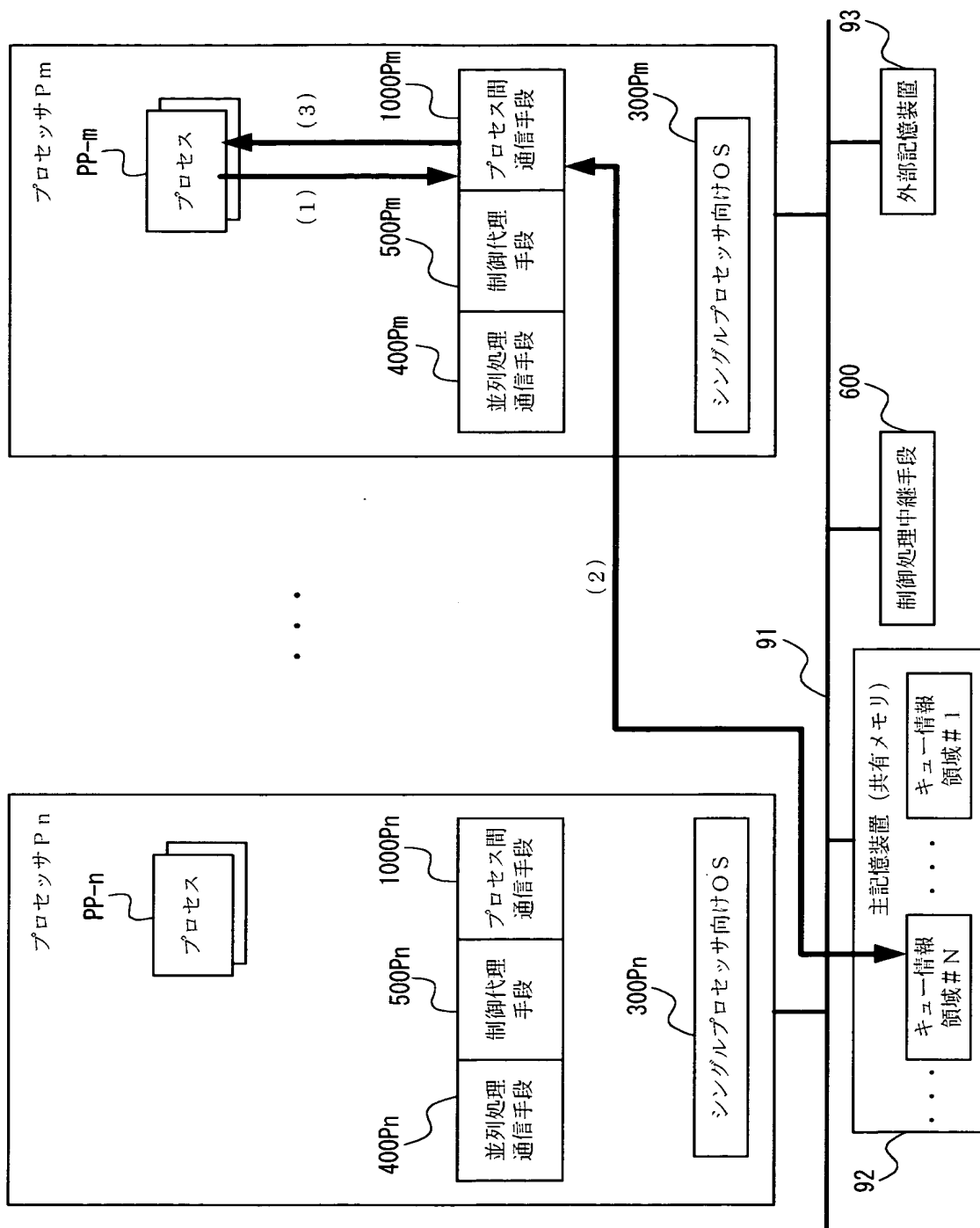
【図 21】

(異なるプロセッサ間のセマフォ方式によるup動作)



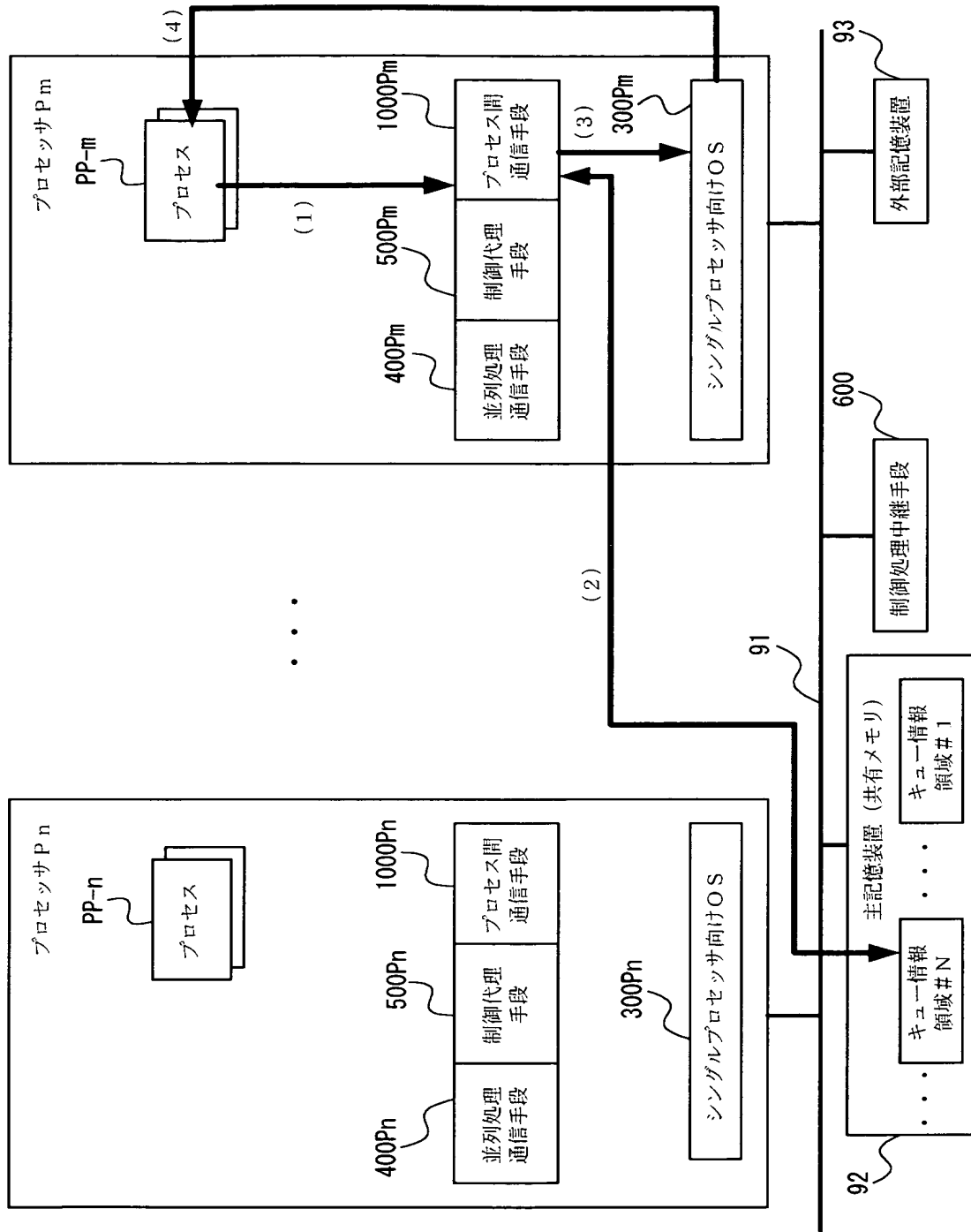
【図 22】

(メッセージキュー方式によるメッセージ受信動作)

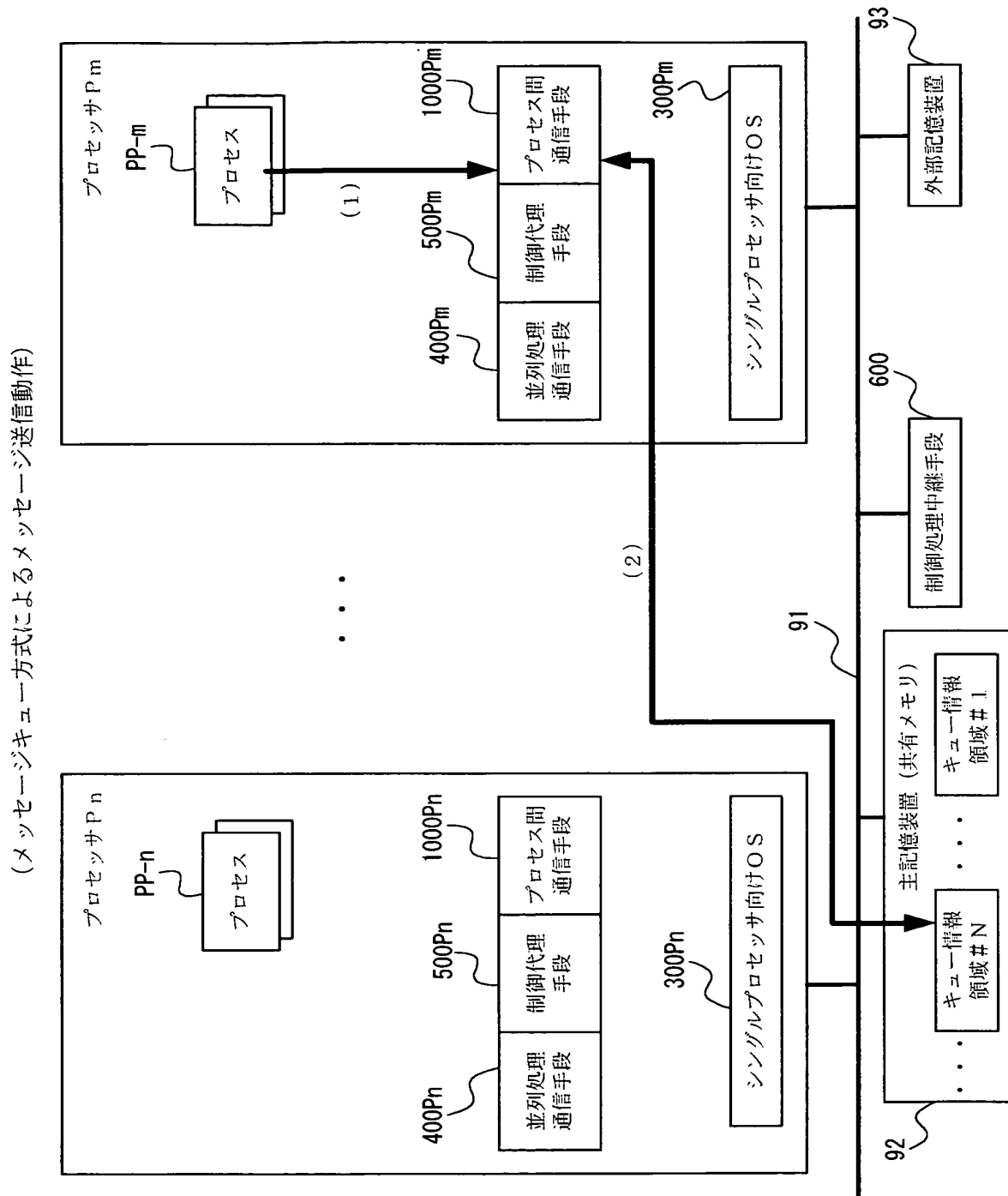


【図 23】

(メッセージキュー方式によるメッセージ受信動作)

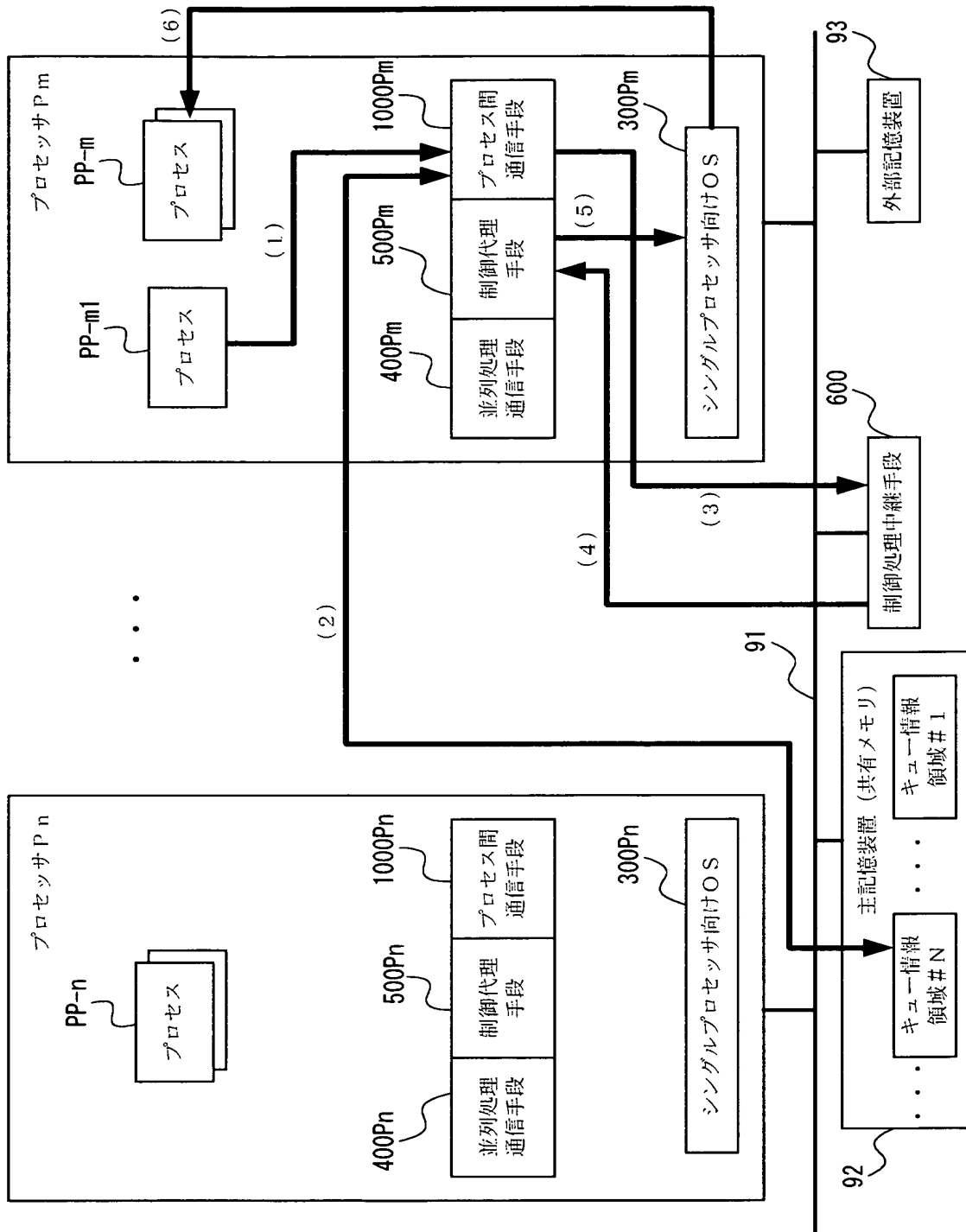


【図 24】

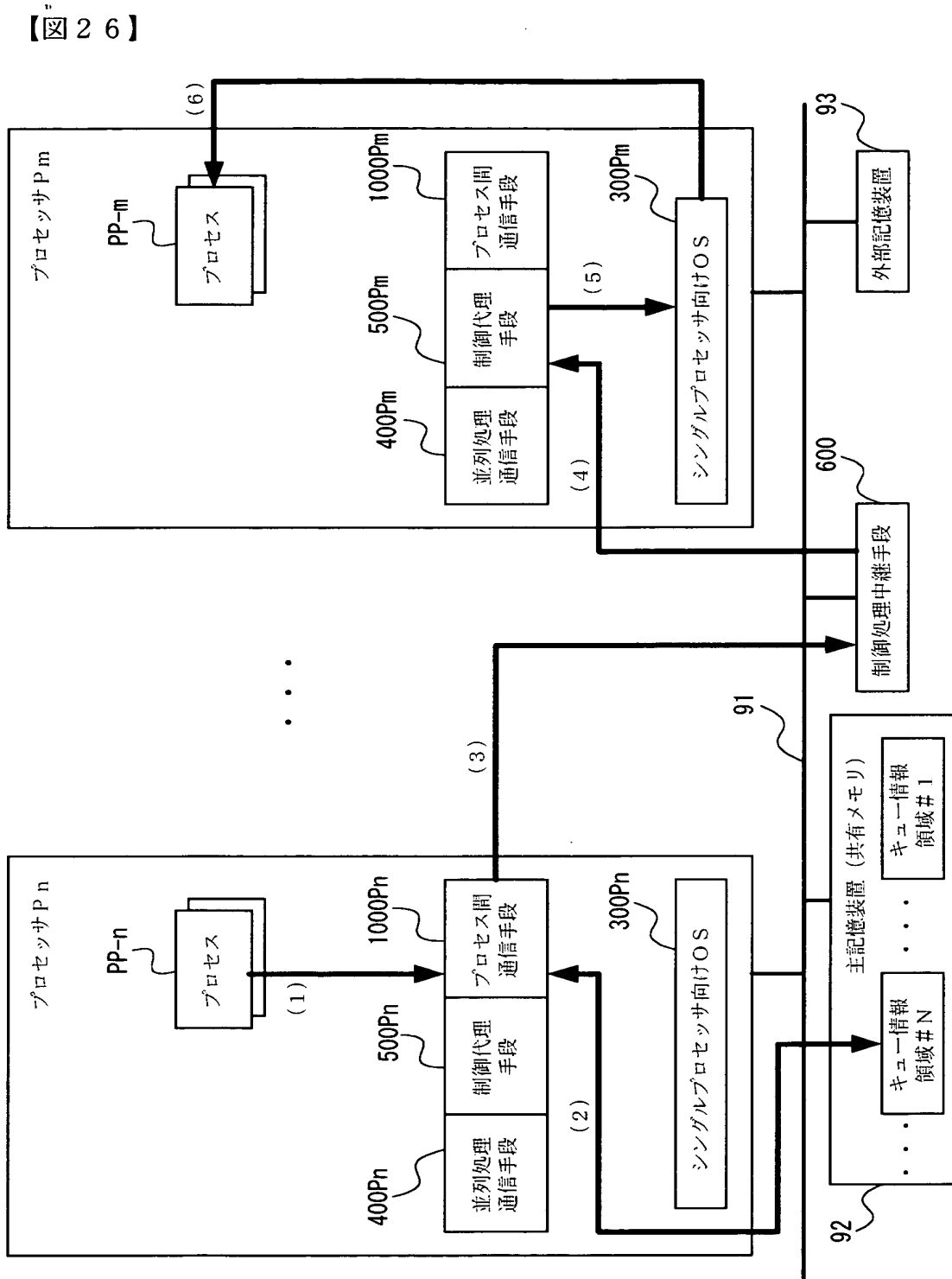


【図 25】

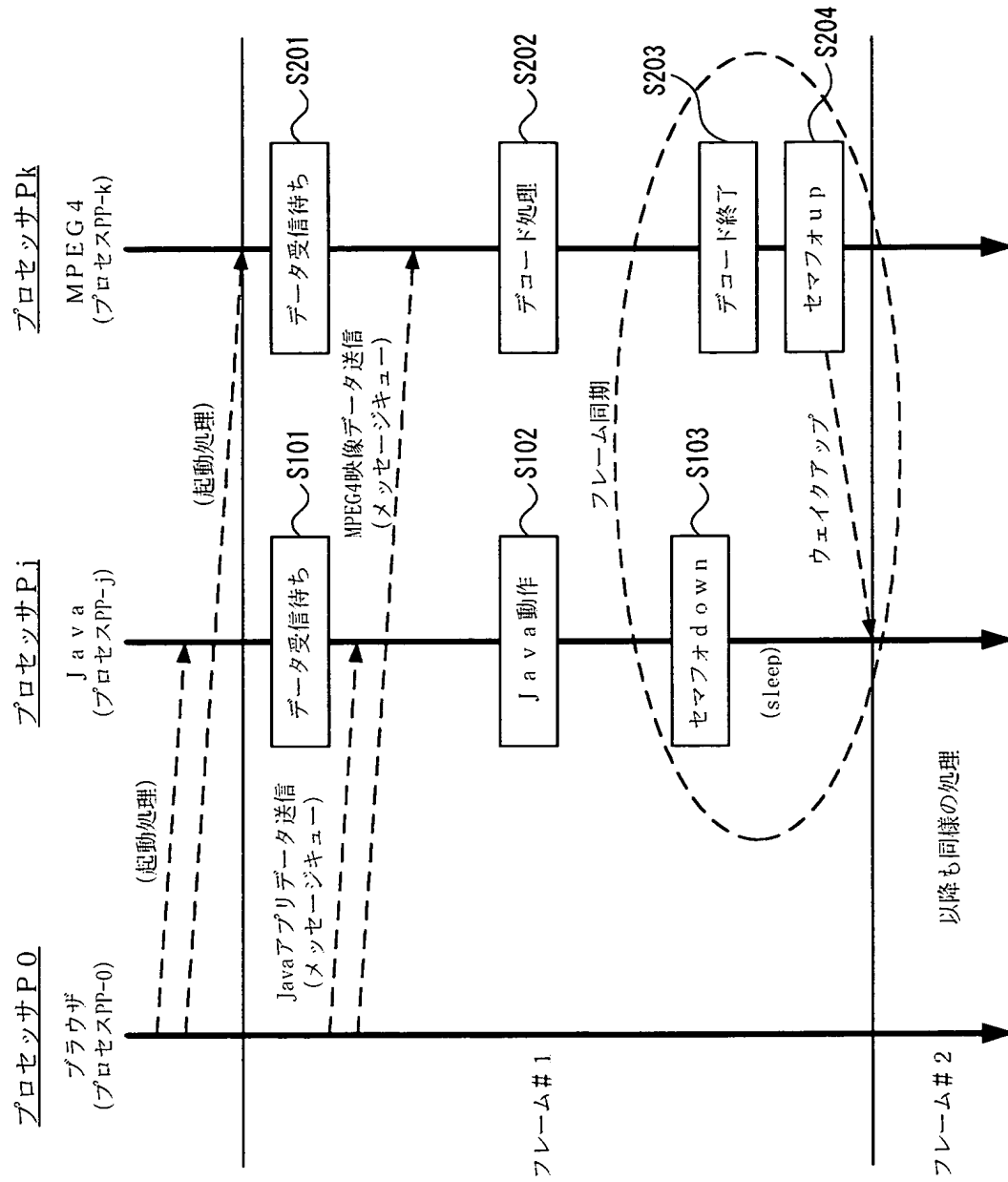
(同一プロセッサ内のメッセージキュー方式によるメッセージ送受信動作)



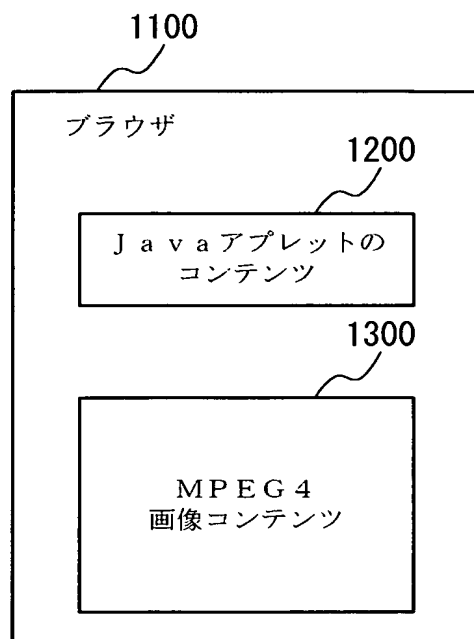
(異なるプロセッサ間のメッセージキュー方式によるメッセージ送信動作)



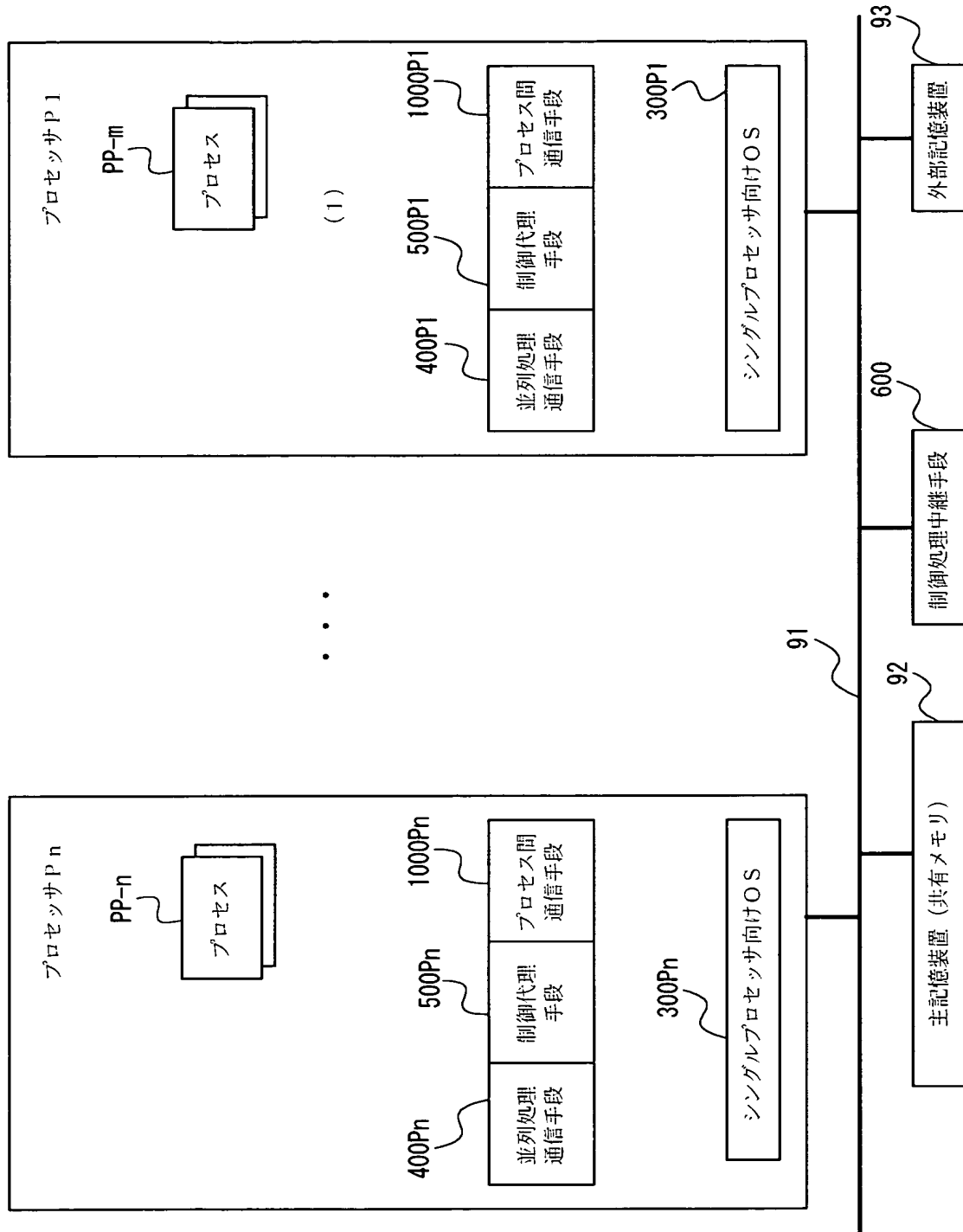
【図 27】



【図 28】



【図 29】



【書類名】 要約書

【要約】

【課題】 マルチプロセッサ上でシングルプロセッサ向けOSと既存のアプリケーションをそれらに改造を加えることなく動作させ、アプリケーションに対してマルチプロセッサによる並列処理を実現できるようにした並列処理システムを提供する。

【解決手段】 マルチプロセッサ上でシングルプロセッサ向けOS及び既存のアプリケーションを動作させ、アプリケーションに対してマルチプロセッサによる並列処理を実現するシングルプロセッサ向けOSによる並列処理システムでは、マルチプロセッサを論理的に、第1のプロセッサ側（10、100）と第2のプロセッサ側（20、200）の二つに分け、第1のプロセッサ側のプロセッサ（P0）上で動作するアプリケーション内で並列化が可能な処理単位（PP）を、第2のプロセッサ側のプロセッサ上で新たな処理単位（PP-1～PP-n）⁹として制御する。

【選択図】 図1

認定・付加情報

特許出願の番号	特願 2 0 0 3 - 0 9 4 7 6 8
受付番号	5 0 3 0 0 5 3 0 4 9 3
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 5 年 4 月 1 日

< 認定情報・付加情報 >

【提出日】 平成15年 3月31日

次頁無

特願 2 0 0 3 - 0 9 4 7 6 8

出 願 人 履 歷 情 報

識別番号

[0 0 0 0 0 4 2 3 7]

1. 変更年月日

1 9 9 0 年 8 月 2 9 日

[変更理由]

新規登録

住 所

東京都港区芝五丁目 7 番 1 号

氏 名

日本電気株式会社